

۱- چکیده.....	۳
۲- مقدمه.....	۵
۳- ۱- نقاط شکست.....	۵
۴- ۱-۱- مفهوم مجموعه نقاط شکست.....	۵
۴- ۱-۲- مزایای استفاده از نقطه شکست(کم کردن تعداد نامعادلات برای حل).....	۱
۴- ۱-۳- تئوری گراف برای پیدا کردن نقاط شکست.....	۱۰
۴- ۲- کارهایی که تا کنون انجام شده است.....	۲۲
۴- ۳- مقدمه‌ای بر هوش مصنوعی.....	۲۳
۴- ۴- سیستمهای خبره.....	۲۴
۴- ۵- مقدمه.....	۲۴
۴- ۶- آینده سیستم‌های خبره.....	۳۱
۴- ۷- کاربرد سیستمهای خبره قدرت.....	۳۲
۴- استفاده از سیستم خبره در پیدا کردن نقاط شکست.....	۳۶
۴- ۱- دلیل استفاده از سیستم خبره در بدست آوردن نقاط شکست.....	۳۶
۴- ۱-۱- در نظر نگرفتن پارامترهای سیستم قدرت در تئوری گراف.....	۳۷
۴- ۱-۲- برابر بودن اعداد در ماتریس _{NRT} و _{NRL}	۳۸
۴- ۱-۳- کار نکردن تئوری گراف به عنوان یک بخش مستقل.....	۴۰
۴- ۱-۴- یکی بودن امتیاز ماتریس _{NRL} و _{NRT}	۴۲
۴- ۱-۵- عوض کردن نقاط شکست در صورت نرسیدن به جواب.....	۴۳
۴- ۲- قوانین سیستم خبره.....	۴۴
۴- ۲-۱- دور بودن و نزدیک بودن به منبع.....	۴۵
۴- ۲-۲- سطح اتصال کوتاه.....	۴۸
۴- ۲-۳- عنصر سریع.....	۴۹
۴- ۳-۴- تعداد رله‌های هم حلقه.....	۵۰
۴- ۳-۵- حلقه با تعداد رله کمتر.....	۵۲

۳-۲-۶- داشتن یا نداشتن حفاظت پایلوت.....	۵۴
۳-۲-۷- تعداد خطوط رسیده به یک شینه.....	۵۶
۳-۲-۸- نوع بارها.....	۵۸
۳-۳- استنتاج.....	۶۰
۳-۴- آزمایش برنامه بر چند شبکه نمونه.....	۶۲
۳-۴-۱) شبکه اول مورد آزمایش:.....	۶۲
۳-۴-۲) شبکه دوم مورد آزمایش:.....	۶۶
۳-۴-۳) شبکه سوم مورد آزمایش:.....	۶۹
۳-۴-۴) شبکه چهارم مورد آزمایش:.....	۷۲
۴- نتیجه‌گیری و پیشنهاد ادامه کار.....	۷۵
۴-۱- نتیجه‌گیری.....	۷۵
۴-۲- پیشنهاد ادامه کار.....	۷۶
۴-۲-۱- اضافه کردن قوانین به صورت تجربی: (نبودن منبع آگاه).....	۷۶
۴-۲-۲- ایجاد یک سیستم خبره واقعی.....	۷۶
۴-۲-۳- وارد کردن جفت جریانهای مختلف.....	۷۷
۴-۲-۴- وارد کردن فیوز در شبکه.....	۷۷
۴-۲-۵- بررسی نقطه ژرف در شبکه‌های رینگبی عنوان یکی از قوانین.	۷۷
۴-۲-۵-۱- امتحان کردن قوانین با هماهنگی رله‌ها.....	۷۸
۵- مراجع.....	۷۹

۱- چکیده

امروزه شبکه‌های قدرت به منظور بالا رفتن پایداری و هدفهای خاص دیگر، به شبکه‌های بزرگ به هم پیوسته تبدیل می‌شوند. این بهم پیوستگی مشکلاتی را در حفاظت شبکه‌های قدرت ایجاد می‌کند. چون در این سیستمها مسئله‌هایی نظیر نقاط شروع هماهنگی و انتخاب پشتیبان خیلی مشکل می‌شود. برای هماهنگی رله‌های شبکه‌های بهم پیوسته راهکارهای زیادی پیشنهاد شده است. از جمله این روش‌ها می‌توان روش‌هایی دوچفت‌جريان، چهارچفت‌جريان، شش‌چفت‌جريان، روش‌های بهینه و ... را نام برد. در تمامی این روش‌ها سعی بر این است که تنظیمات زمانی و جریانی رله‌ها را برای حالتی که تمامی رله‌ها هماهنگ باشند بدمت آورند. ولی در تمامی این روش‌ها یک نکته کلی مشترک است. در حقیقت اساس تمامی این روش‌ها بر پیدا کردن یک مجموعه رله برای شروع هماهنگی است، به طوریکه وقتی رله‌های این مجموعه هماهنگ شد، هماهنگی باقی رله‌ها با موفقیت صورت بگیرد. یکی از مهمترین خاصیتها این مجموعه ایست که با باز کردن شبکه در راستای حفاظتی رله‌های این مجموعه، دیگر حلقه‌ای در شبکه باقی نمی‌ماند. در نتیجه انتخاب پشتیبان برای رله‌ها بسیار آسان است. پس با انتخاب این نقاط دو مشکلی که در هماهنگی رله‌های شبکه‌های بهم پیوسته بود بر طرف می‌شود.

مهمترین و اساسی‌ترین روش موجود برای پیداکردن نقاط شکست در سیستم‌های بهم پیوسته، روش تئوری گراف است [۸و۷]. در تئوری گراف، تنها مسئله اصلی که به آن پرداخته شده است، توپولوژی سیستم قدرت در یافتن نقاط شروع هماهنگی و مجموعه رله‌ها است. یعنی نکات مورد توجه در این روش، خطوط و باسهای می‌باشند و حتی به مسائلی همچون محل منابع تغذیه نسبت به نقاط شروع و یا میزان سطوح اتصال کوتاه پرداخته نمی‌شود. با یک مقایسه ساده در شبکه‌های شعاعی مشخص است که این مسئله که منبع در کجای سیستم باشد و سطوح اتصال کوتاه چقدر باشند در تعیین نقطه شروع کار مهم است. پس در شبکه‌های بهم پیوسته، بعلت پیچیدگی شبکه و تنوع حفاظت علاوه بر موارد یاد شده، مسائل دیگری نیز باشیستی منظور شود. در راستای این اشکالات با مد نظر داشتن تئوری گراف، پارامترهای دیگری را که در تئوری گراف و تمامی روش‌های پیشین وارد نشده است در نظر گرفته می‌شود. این پارامترها با استفاده از تجربیات اشخاص خبره عنوان شده‌اند و با گردآوری آنها و تبدیل آنها

به صورت قوانین سیستم خبره و اعمال آنها به برنامه کامپیوتری جوابهای کاملاً متفاوت و بسیار معقولتر از تئوری گراف گرفته شد.

نکته قابل توجه در این طرح پیشنهادی اینست که تئوری گراف یکی از قوانین سیستم خبره پیشنهاد شده است. در حقیقت تئوری گراف و چندین قانون دیگر به کمک هم بهترین نقاط شکست را می‌دهد. این قوانین در داخل پروژه توضیح داده می‌شوند و طرز اعمال آنها به برنامه کامپیوتری، شرح داده خواهد شد.

در این پروژه در فصل اول به توضیح پیرامون مفهوم نقاط شکست پرداخته می‌شود. با توجه به گفته‌های بالا، از آنجا که تئوری گراف اساس روشهای موجود است، در فصل دوم به توضیح و تشریح تئوری گراف بدون توجه به اثباتهای ریاضی آن پرداخته می‌شود. در فصل سوم به توضیح پیرامون سیستم خبره و کاربردهای آن در بخش قدرت پرداخته می‌شود. در بخش بعدی که فصل اصلی پروژه است، به توضیح قوانین سیستم خبره می‌پردازیم. این قوانین کامل کننده تئوری گراف در بدست آوردن نقاط شکست در سیستمهای بهم پیوسته است. در این فصل برای بهتر مشخص شدن نتایج، یک مثال همراه قوانین آورده شده است.

۲- مقدمه

۱-۱- نقاط شکست

۱-۱-۱- مفهوم مجموعه نقاط شکست

در این قسمت به این امر می‌پردازیم که مجموعه نقاط شکست چه مجموعه‌ای هستند.

در حقیقت مفهوم نقاط شکست در این مبحث عنوان می‌شود.

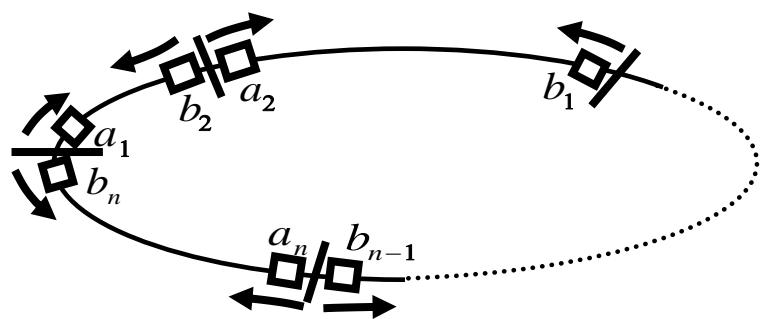
در بحث مفهوم نقاط شکست به دو مسأله اساسی پرداخته می‌شود. این دو مسأله

عبارتند از:

اولاًً اینکه اساساً چرا از نقاط شکست استفاده می‌شود و اینکه اگر مجموعه نقاط شکست استفاده نشوند چه مشکلاتی در هماهنگی رله‌ها پیش می‌آید. پس ضرورت استفاده از نقاط شکست در این قسمت بیان می‌گردد.

ثانیاً اینکه طریقه هماهنگی رله‌ها با استفاده از نقاط شکست چگونه است. یعنی الگوریتم کلی که هماهنگی رله‌ها را با استفاده از نقاط شکست عنوان کند، معرفی گردد. البته منظور این پروژه بررسی و معرفی الگوریتم‌های هماهنگی نیست. بلکه توضیح می‌دهیم که نقاط شکست چگونه در ابتدای هماهنگی، کار هماهنگ کردن را ساده‌تر می‌کنند.

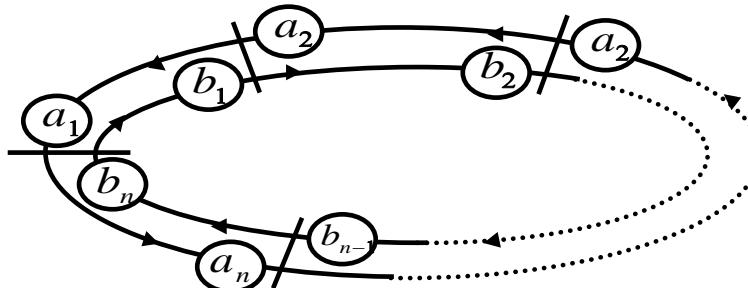
برای تشریح مفهوم مجموعه نقاط شکست، هماهنگی حفاظت یک سیستم در شکل (۱) به عنوان یک مثال داده می‌شود. در این شبکه تک حلقه‌ای، هر رله جهت دار، تابع پشتیبان رله باس دور را کنترل می‌کند (یعنی a_1 پشتیبان a_2 است، یا بطور مساوی a_2 رله اصلی برای a_1 است). برای دستیابی به حفاظت هماهنگ شده، هر رله باید بر اساس تنظیمات رله اصلی خودش تنظیم شود. بنابراین تنظیم رله a_2 وابسته به تنظیمات رله a_1 است. بطور مشابه، تنظیمات $a_1, a_2, a_3, \dots, a_n$ بر اساس وابستگی‌های پشتیبان-اصلی در میان خودشان متوالیاً به یکدیگر وابسته هستند. در همان راستا، تنظیمات $b_1, b_2, b_3, \dots, b_n$ متوالیاً به یکدیگر وابسته هستند.



شکل ۱ - دیاگرام تک خطی مدار قدرت

این وابستگی در میان رله‌ها می‌تواند توسط شکل (۲) نشان داده شود، که به اصطلاح نمودار وابستگیها نامیده می‌شود. در این دیاگرام، خطوط جهت‌دار ارتباط پشتیبان-اصلی میان رله‌ها را مشخص می‌کنند. به عنوان یک مثال، خط جهت‌دار از a_2 به a_1 مشخص می‌کند که رله اصلی a_1 است.

شکل ۲ - دیاگرام تک خطی مدار قدرت



برای هماهنگی این حفاظت، یک مجموعه مناسب از رله‌ها، به نام مجموعه نقاط شکست، باید به عنوان نقطه شروع روش هماهنگی مشخص شوند. یک می‌نیمم مجموعه نقاط شکست، یک تعداد می‌نیمم از رله‌ها است، شامل حداقل تعداد رله‌ها در هر حلقه ساعتگرد و پادساعتگرد. بر طبق دیاگرام وابستگی شکل ۲، یک می‌نیمم برای حفاظت سیستم نمونه باید شامل باشد یک رله از a_i و دیگری یک رله از b_i ، مثلاً $\{a_2, b_1\}$.

تأثیر انتخاب یک رله بعنوان یک نقطه شکست می‌تواند روی نمودار وابستگی، با حذف همه جهت‌هایی که به آن رله می‌آیند نشان داده شود. برای مثال اثر انتخاب a_1 بعنوان یک نقطه شکست می‌تواند روی نمودار وابستگی در شکل (۲) توسط حذف جهت a_1a_2 اعمال شود. به طور کلی تعیین می‌نیمم مجموعه نقاط شکست در یک شبکه چند حلقه‌ای مشخص کردن یک تعداد می‌نیمم از رله‌ها روی خطوط شبکه است. چنین مجموعه‌ای از رله‌های جهت دار، باید همه مسیر حلقه‌ها را در هر دو جهت ساعتگرد و پادساعتگرد باز کند.

با این توضیحات، مفهوم اولیه مجموعه نقاط شکست گفته شد و معلوم شد که مجموعه نقاط شکست، مجموعه رلهایی هستند که همه حلقه‌های شبکه را در دو جهت ساعتگرد و پاد ساعتگرد باز می‌کند. در حقیقت وقتی رله‌ای نقطه شکست است، شبکه را در جهت دید خود باز می‌کند. اشتباهی که عمدتاً در مورد نقطه شکست عنوان می‌شود اینست که می‌گویند وقتی رله‌ای نقطه شکست است، شبکه از آنجا باز می‌شود. ولی این اشتباه را می‌توان اینگونه تصحیح کرد که شبکه در جهت دید رله باز می‌شود و در جهت دیگر برقرار است. ولی اگر دو رله در دو طرف خط نقطه شکست شوند، آنوقت می‌توان گفت که شبکه از آن خط کاملاً باز می‌شود و خط از شبکه خارج می‌شود.

۱-۲-۳- مزایای استفاده از نقطه شکست (کم کردن تعداد نامعادلات

(برای حل)

با توجه به مطالب گفته شده تا کنون دیده شد که مجموعه نقاط شکست به عنوان شروع کننده هماهنگی، می‌تواند ترتیب رله‌ها را معلوم کند. از جهتی در هماهنگی رله‌ها روشهایی وجود دارد که می‌تواند بدون استفاده از مجموعه نقاط شکست، عمل هماهنگی را انجام داد. در این قسمت به توضیح این مسأله می‌پردازیم که نقاط شکست برای آنگونه راه حلها چه مزیتی را به ارمغان می‌آورد و چه ساده‌سازیها و یا چه امکاناتی در آن راه حلها فراهم می‌آورد.

مهمنترین مزیت استفاده از نقطه شکست در حفاظت سیستمهای قدرت کم کردن تعداد نامعادلات برای حل دستگاه‌های غیر خطی می‌باشد.

در روشهای ابتدایی هماهنگی در شبکه‌های بهم پیوسته، ماتریسی تشکیل می‌شود که در آن تمامی پشتیبانهای هر رله مشخص است. در این ماتریس مشخص می‌شود رله‌های پشتیبان هر رله کدام است. یکی از نکات دیگری که در این ماتریس مشخص می‌شود اینست که اگر تنظیمات رله‌ای معلوم باشد، تنظیمات کدام رله‌ها معلوم می‌شود. پس یک ماتریسی موجود است که وابستگی‌های بین رله‌ها را به صورت یک شبکه پیچیده و تودرتو معلوم می‌کند.

در مرحله بعد، با داشتن این اطلاعات، شرائط هماهنگی برای همه رله‌ها نوشته می‌شود. این شرائط می‌تواند عمل کردن رله اصلی قبل از رله پشتیبان و عمل نکردن رله پشتیبان قبل از رله اصلی و ... باشد. حال یک دستگاه نامعادلات بسیار حجمی حاصل می‌شود. با توجه به اینکه دستگاه نامعادلات حاصل، یک دستگاه نامعادلات غیر خطی است و تعداد مجهولات آن نیز بسیار زیاد است، حل آن کاری وقت‌گیر و در بعضی موارد بدون نتیجه است. برای حل چنین مشکلی مجموعه نقاط شکست تا حد زیادی می‌توانند کمک کنند.

اینکار اینگونه انجام می‌شود که اول تعداد وابستگی‌های بین رله‌ها را کم می‌کنیم. یعنی رله‌هایی را به عنوان پایه انتخاب می‌کنیم و باقی رله‌ها را بر اساس آنها تنظیم می‌کنیم و یا در حالت کلی تر اگر ماتریس وابستگی‌ها را با توجه به این رله‌های پایه بنویسیم، میرهن است که ماتریس حاصل دارای وابستگی‌های کمتری در اعضاش است. ولی مسأله‌ای که باقی می‌ماند اینست که پس از هماهنگی رله‌های شبکه، خود این رله‌ها باید با رله‌های اصلی خود هماهنگ

باشند. پس در آخر هماهنگی، باید هماهنگی این رله‌ها با تمام رله‌های اصلی خودشان چک شود. در این مرحله دو حالت ممکن است ایجاد شود. اول اینکه هماهنگی بین همین رله‌ها برقرار باشد. مشخص است که این حالت انتهایی هماهنگی را مشخص می‌کند. دومین حالت اینست که به ناهمانگی بین رله‌های مجموعه نقاط شکست و رله‌های اصلی رسیده شود. در این مرحله عموماً دو راه حل زیر پیشنهاد می‌شود. راه اول اینست که کار انتخاب مجموعه نقاط شکست دوباره انجام شود. یعنی یک سری رله جدید به عنوان مجموعه نقاط شکست انتخاب شود. سپس کار هماهنگی از اول النجام شود و در آخر هماهنگی بین رله‌های مجموعه نقاط شکست با رله‌های اصلی خودشان چک شود. دومین راه حل اینست که مسیر بین رله نقطه شکست و رله اصلی این رله که ناهمانگی از آن رله آمده است را با یک حافظت دیگر مثل حافظتهای واحد حفاظت کرد و کاری به هماهنگی بین رله نقطه شکست و رله اصلی آن نداشت.

با انجام یکی از کارهای بالا سیستم هماهنگ می‌شود.

نتیجه بحث این است که برای هماهنگی خوب و سریع بین رله‌ها بهترین کار اینست که یک مجموعه از رله‌ها به عنوان مجموعه نقاط شکست معرفی بشوند. سپس با باز کردن شبکه در جهت این رله‌ها به شبکه ساده‌تری می‌رسیم. با تنظیم رله‌های این شبکه که بسیار راحت‌تر و سریع‌تر از تنظیم رله‌های شبکه اصلی است و در آخر با اعمال یکی از دو روش برای ناهمانگی‌ها، رله‌های شبکه هماهنگ می‌شود.

۱-۳-۲- تئوری گراف برای پیدا کردن نقاط شکست

در این قسمت به توضیح تئوری گراف بدون توجه به بازکردن مسائل دقیق ریاضی در مورد این تئوری می‌پردازیم. دلایل توضیح تئوری گراف در زیر عنوان می‌شود:

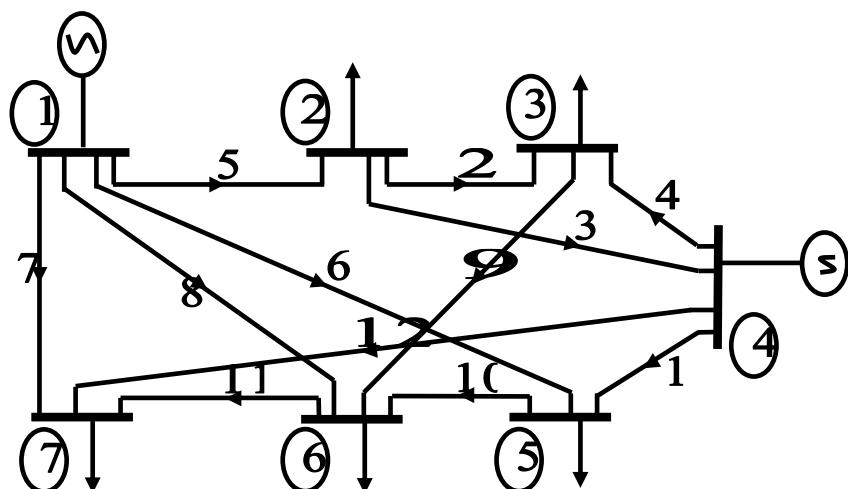
۱. از آنجا که بیشتر راه حل‌های پیشنهاد شده برای پیدا کردن نقاط شکست بر مبنای تئوری گراف می‌باشد، آشنایی با این تئوری می‌تواند اشراف شخص را بر موضوع مورد بحث بیشتر کند.

۲. یکی از احتیاجات دیگر به تئوری گراف، باز کردن مشکلات موجود در راه حل‌های دیگر است. در حقیقت بدون آشنایی با ماتریس‌های تئوری گراف، نمی‌توان به بحث در پیرامون روش‌های موجود و اشکالات آن پرداخت.

۳. چون در آخر، در روش پیشنهادی این پروژه، از تئوری گراف در سیستم خبره استفاده شده است، توضیح این قسمت ضروری به نظر می‌آید.

مراحل تئوری گراف

در این مرحله به تشریح راه حل تئوری گراف در پیدا کردن نقاط شکست پرداخته می شود. برای بهتر مفهوم شدن این مرحله، در طی توضیح این مراحل، یک مثال از یک شبکه نمونه، آورده شده است. گراف این نمونه در شکل (۳) نشان داده شده است. این شبکه دارای ۷ بس و ۱۲ شاخه است.



شکل (۳)- نمونه سیستم قدرت بهم پیوسته

برای اینکار با استفاده از تئوری گراف حلقه های گراف بدست آورده می شود. سپس کلیه حلقه های شبکه پیدا شده و حلقه های ساده و چندگانه بدست آورده می شود. از روی این حلقه ها، حلقه های ساده را جدا می کنیم. سپس حلقه ها را جهت دار می کنیم و تعداد رله ها در هر حلقه و تعداد رله های پشتیبان هر رله بدست می آید. سپس با استفاده از بعضی قوانینی که عنوان می شود، به تعیین اولین نقطه شکست می پردازیم. با بدست آوردن اولین نقطه شکست و باز کردن شبکه در آن جهت و تکرار بعضی از قسمتهای باقی نقاط شکست بدست می آیند.

تشکیل ماتریس تلاقي

در تئوری گراف، شبکه قدرت به صورت یک گراف با یک سری گره و شاخه مشخص می‌گردد. پارامترهای دیگر در این تئوری مهم نیستند. پس می‌توان ورودی تئوری گراف را ماتریس تلاقی گراف دانست. تعداد سطرهای ماتریس تلاقی برابر تعداد گره‌های شبکه و تعداد ستونهای این ماتریس برابر تعداد شاخه‌های شبکه است. درایه‌های این ماتریس با فرمول زیر بدست می‌آیند:

$$\hat{A}_{ij} = \begin{cases} 1 & \text{اگر خط زام از گره } i \text{ خارج شود} \\ 0 & \text{اگر خط زام با گره } j \text{ تلاقی نکند} \\ -1 & \text{اگر خط زام به گره } j \text{ فوارد شود} \end{cases}$$

پس متناظر با هر گرافی یک ماتریس تلاقی تعریف می‌کنیم که درایه‌های آن اعداد (۰ یا ۱) است. اگر n تعداد گره‌های سیستم قدرت و e تعداد خطوط شبکه باشد، ماتریس تلاقی بصورت زیر خواهد بود:

ماتریس تلاقی	خطهای شبکه
n	\hat{A}

در مورد مثال شکل (۳)، ماتریس تلاقی بصورت زیر خواهد بود:

ماتریس تلاقی	خطوط											
	1	2	3	4	5	6	7	8	9	۰۱	۱۱	۲۱
۷	1	0	0	0	0	1	1	1	1	0	0	0
	2	0	1	1	0	-1	0	0	0	0	0	0
	3	0	-1	0	-1	0	0	0	0	1	0	0
	4	1	0	-1	1	0	0	0	0	0	0	1
	5	-1	0	0	0	0	-1	0	0	0	1	0
	6	0	0	0	0	0	0	0	-1	-1	-1	0
	7	0	0	0	0	0	0	-1	0	0	0	-1

اگر گره شماره آخر را به عنوان مرجع انتخاب کنیم، ماتریس به صورت مختصر شده در می‌آید. در این ماتریس سطر آخر حذف می‌شود. نام این ماتریس ماتریس تلاقی مختصر شده است. برای شکل شماره (۳) داریم که:

		خطوط												
		1	2	3	4	5	6	7	8	9	01	11	21	
مختصر شده	ماتریس تلاقي	1	0	0	0	0	1	1	1	1	0	0	0	0
		2	0	1	1	0	-1	0	0	0	0	0	0	0
		3	0	-1	0	-1	0	0	0	0	1	0	0	0
		4	1	0	-1	1	0	0	0	0	0	0	0	1
		5	-1	0	0	0	0	-1	0	0	0	1	0	0
		6	0	0	0	0	0	0	-1	-1	-1	1	1	0

بدست آوردن درخت گراف

در این مرحله درخت شبکه تشکیل داده می‌شود. بدست آوردن درخت یک گراف بصورت الگوریتم کامپیوتری را می‌توان بصورت زیر بنا کرد که یک نقطه دلخواه در نظر گرفته می‌شود و سپس تمام گره‌هایی را که با این گره در رابطه هستند از روی ماتریس تلاقي بدست آورده می‌شوند و هر کدام از این گره‌ها از مجموعه کل گره‌ها حذف می‌شوند. هر گره باید با یکی از گره‌های مجموعه حاصل در رابطه باشد. اگر اینگونه نباشد دو گراف مجزا وجود داشته و چون دو گراف مجزا برای هماهنگی بهم مربوط نیستند، این حالت بحث نمی‌شود. حال لینکها و شاخه‌ها شماره گذاری می‌شوند. سپس ماتریس تلاقي جدید نوشته می‌شود. برای شکل (۳) این ماتریس بصورت زیر در می‌آید:

		خطوط												
		5	6	8	2	4	5	7	3	9	01	11	21	
ماتریس تلاقي	ماتریس تلاقي	1	1	1	1	0	0	0	1	0	0	0	0	0
		2	-1	0	0	1	0	0	0	1	0	0	0	0
		3	0	0	0	-1	-1	0	0	0	1	0	0	0
		4	0	0	0	0	1	1	0	-1	0	0	0	1
		5	0	-1	0	0	0	-1	0	0	0	1	0	0
		6	0	0	-1	0	0	0	0	0	-1	-1	1	0
		7	0	0	0	0	0	0	-1	0	0	0	-1	-1

گره مرجع، گره شماره آخر است. اگر سطر آخر را حذف کنیم ماتریس A بدست می‌آید که ماتریس تلاقي خلاصه شده نامیده می‌شود. این ماتریس برای شکل (۳) بصورت زیر است:

ماتریس تلاقي	خطوط												
	5	6	8	2	4	5	7	3	9	01	11	21	
$\hat{A} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$	1	1	1	1	0	0	0	1	0	0	0	0	0
	2	-1	0	0	1	0	0	0	1	0	0	0	0
	3	0	0	0	-1	-1	0	0	0	1	0	0	0
	4	0	0	0	0	1	1	0	-1	0	0	0	1
	5	0	-1	0	0	0	-1	0	0	0	1	0	0
	6	0	0	-1	0	0	0	0	0	-1	-1	1	0

حال ماتریس A را به دو بخش تقسیم می‌کنیم. اگر l تعداد لینکها و b تعداد شاخه‌ها باشد، ماتریس A را بصورت زیر تبدیل می‌کنیم:

A	l لینکها	b شاخه‌ها
$n - 1$	A_l	A_b

در مورد شکل (۳) این ماتریس بصورت زیر درمی‌آید.

ماتریس تلاقي	خطوط												
	5	6	8	2	4	5	7	3	9	01	11	21	
	لینکها												
$\hat{A} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}$	1	1	1	1	0	0	0	1	0	0	0	0	0
	2	-1	0	0	1	0	0	0	1	0	0	0	0
	3	0	0	0	-1	-1	0	0	0	1	0	0	0
	4	0	0	0	0	1	1	0	-1	0	0	0	1
	5	0	-1	0	0	0	-1	0	0	0	1	0	0
	6	0	0	-1	0	0	0	0	0	-1	-1	1	0
	7	0	0	0	0	0	0	-1	0	0	0	-1	-1

ماتریس حلقه‌های اساسی

حال ماتریس حلقه‌های اساسی پیدا می‌شوند. جهت حلقه، جهت لینک و تعداد حلقه، تعداد لینک است. حال ماتریس L_B بصورت زیر تشکیل می‌شود:

L_B	l لینکها	b شاخه‌ها
-------	------------	-------------

پیدا کردن نقاط شکست

۱۵

حلقه‌های اساسی	I_l	L_b
----------------	-------	-------

که در ماتریس بالا L_b با فرمول زیر حساب می‌شود:

$$L_b = (-A^{-1} \times A_l)^T$$

درمورد مثال شکل (۳) این ماتریس بصورت زیر است:

L_B	لينکها: l						شاخه‌ها: b					
	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
حلقه‌های اساسی	۱	۱	۰	۰	۰	۰	۰	-۱	۱	۰	۰	۰
	۲	۰	۱	۰	۰	۰	۰	-۱	۰	۰	۱	۱
	۳	۰	۰	۱	۰	۰	۰	-۱	۰	۰	۰	۰
	۴	۰	۰	۰	۱	۰	۰	۰	-۱	۱	۰	-۱
	۵	۰	۰	۰	۰	۱	۰	۰	۰	۱	۰	-۱
	۶	۰	۰	۰	۰	۰	۱	۰	۰	۰	۱	-۱

حال ماتریس کلیه حلقه‌های شبکه تشکیل می‌شود. برای این کار اول \bar{L}_B تشکیل می‌شود. \bar{L}_B همان L_B است، با این تفاوت که علامتهای منفی به مثبت تبدیل می‌شود. سپس ماتریس E به صورت زیر بدست می‌آید:

E	l
l	مقدار باینری عدد 2^{l-1}
	مقدار باینری عدد 2^{l-2}
	\vdots
	مقدار باینری عدد ۱
$2^{l-1}-l-1$	مقدار باینری عدد ۳
	مقدار باینری عدد ۵
	\vdots
	مقدار باینری عدد $2^l - 1$

حال ماتریس E به دو ماتریس بصورت زیر تبدیل می‌شود:

E	لینکها: l
l	I_l
$2^l - l - 1$	\hat{E}

سپس ماتریس C به صورت زیر تعریف می‌شود:

C	لینکها: l	شاخهها: b
l	I_l	$\overline{L_b}$
$2^l - l - 1$	\hat{E}	$\hat{E} \times \overline{L_b}$

این ماتریس شامل همه حلقه‌های شبکه اعم از ساده و چندگانه است. منظور از حلقه چندگانه حلقه‌هایی است که از بهم چسبیدن دو حلقه بدست می‌آیند.

مشخص کردن حلقه‌های چندگانه و حذف آنها

حال ماتریس تعداد شاخه‌های حلقه‌ها بدست می‌آیند. این ماتریس از فرمول زیر بدست می‌آیند:

$$D = C * C^T$$

در مورد مثال شکل (۳)، ماتریس D یک ماتریس 63×63 می‌شود که نشان دادن آن در اینجا مورد لزوم نیست.

حال اگر در سطر i ام، عنصر قطری D_{ii} با عنصر D_{ij} برابر بود، حلقه شماره j حلقه چندگانه است. پس سطر j ام را از ماتریس E حذف کرده و ماتریس حاصل را E_s می‌نامیم و ماتریس C_s از فرمول زیر بدست می‌آوریم:

$$C_s = E_s * \overline{L_b}$$

جهت‌دار کردن حلقه‌ها

پس از حذف حلقه‌های چندگانه، باید حلقه‌ها را جهت‌دار کرد. برای اینکار حالت‌های مختلف هر سطر، از نظر مثبت و منفی را نوشت و آن را E'_s می‌نامیم و حاصل عبارت $E'_s * L_b$ را حساب می‌کنیم. اگر در سطری، عدد ۲ یا -۲ مشاهده شد، آن سطر را از E'_s حذف می‌کنیم تا ماتریس \hat{E}_{sd} حاصل شود. سپس ماتریس C_{sd} را از فرمول زیر بدست می‌آوریم:

C	لبنکها: I	شاخه‌ها: b
l	I_l	L_b
	\hat{E}_{sd}	$\hat{E}_{sd} \times L_b$

در این مرحله ماتریس تلاقی حلقه‌ها با رله‌ها را تشکیل می‌دهیم. این ماتریس به صورت زیر بدست می‌آید:

L_D	رله‌ها در جهت مثبت	رله‌ها در جهت منفی
حلقه‌ها در جهت مثبت	L_1	L_2
حلقه‌ها در جهت منفی	L_2	L_1

که در ماتریس بالا L_1 همان ماتریس C_{sd} است، با این تفاوت که همه اعداد ۱- را به صفر تبدیل می‌کنیم و L_2 همان ماتریس $-C_{sd}$ است، با این تفاوت که همه اعداد ۱- را به صفر تبدیل می‌کنیم.

تا این مرحله در حقیقت کار تئوری گراف به انتهای می‌رسد و باید به نتیجه‌گیری از ماتریسهای حاصل از گراف برای پیدا کردن نقطه شکست پرداخت.

پیدا کردن مجموعه نقاط شکست از روی نتایج گراف

تا کنون به معرفی یکسری از ماتریسهای گراف پرداخته شده است. حال باید با این ماتریسهای مجموعه نقاط شکست یک شبکه را بدست آورد.

بنابرآنچه که گفته شد، مجموعه نقاط شکست، یکسری رله هستند که تمامی حلقه‌های شبکه را در دوسوی ساعتگرد و پادساعتگرد باز می‌کنند. پس اگر بتوان یک مجموعه رله با این خاصیت پیدا کرد، می‌توان عنوان مجموعه نقاط شکست را به آن مجموعه داد.

پس الگوریتم بدست آوردن نقاط شکست را میتوان به صورت زیر عنوان کرد:

۱. یک رله به عنوان اولین نقطه شکست معرفی می‌گردد.
۲. تمامی حلقه‌هایی که این رله در آنها قرار دارد باز می‌شوند. (اینکار به این خاطر انجام می‌شود که وقتی یکی از اعضای حلقه‌ای را حذف کنیم کل حلقه از بین می‌رود). برای انجام اینکار، کلیه سطرهای ماتریس L_D که رله مزبور در آن قرار دارد را مساوی صفر قرار می‌دهیم.
۳. اگر ماتریس L_D صفر شد کار به اتمام رسیده است (صفر شدن ماتریس L_D یعنی اینکه دیگر حلقه‌ای در شبکه باقی نمانده است). در غیر اینصورت یک رله دیگر به عنوان دومین نقطه شکست انتخاب می‌گردد.

پیدا کردن مجموعه نقاط شکست بهینه از روی نتایج گراف

الگوریتم گفته شده در بالا یک الگوریتم کلی برای انتخاب مجموعه نقاط شکست است. ولی در این الگوریتم دیده شد که تعداد نقطه شکست مد نظر نبود. اگر تعداد نقاط شکست برای یک شبکه قدرت زیاد شود، مشکلی برای هماهنگی رله‌ها ایجاد می‌شود. این مشکل از آنجا ناشی می‌شود که پس از پیدا کردن نقاط شکست و تنظیم هماهنگی باید هماهنگی بین رله‌های نقطه شکست و باقی رله‌ها کنترل شود. در صورت ناهمانگی، روش‌هایی را که در بخش (۲-۲) عنوان شد باید پیاده شوند. هر چه تعداد رله‌های نقاط شکست کمتر باشد، احتمال وقوع ناهمانگی در مرحله کنترل هماهنگی‌ها کمتر خواهد بود. پس فقط پیدا کردن نقاط شکست مهم نسیت، بلکه کمترین تعداد عضو برای نقطه شکست دارای اهمیت خاصی است. برای نیل به این مقصود احتیاج به معرفی دو ماتریس دیگر می‌باشد که در ذیل آورده شده است.

دو ماتریس NRL و NRT را به صورت زیر تشکیل می‌دهیم که NRL مجموع هر سطر ماتریس L_D است و NRT از رابطه زیر بدست می‌اید :

$$NRT = NRL * L_D$$

در حقیقت ماتریس NRL تعداد رله‌های موجود در هر حلقه را نمایش میدهد و ماتریس NRT نشان می‌دهد که هر رله با چند رله هم حلقه است. یعنی میگوید که با تنظیم یک رله، تنظیمات چند رله که در حلقه‌های شامل رله مذبور هستند معلوم می‌شود.

در مورد مثال شکل (۳) ماتریس NRL بصورت زیر است:

شماره حلقه	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰
تعداد رله در حلقه	۴	۴	۳	۵	۴	۴	۴	۵	۳	۵	۵	۶	۴	۶	۶	۷	۷	۳	۵	۵
شماره حلقه	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱	۳۲	۳۳	۳۴	۳۵	۳۶	۳۷	۳۸	۳۹	۴۰
تعداد رله در حلقه	۶	۶	۶	۵	۷	۵	۷	۵	۵	۵	۴	۶	۶	۶	۴	۶	۵	۷	۷	۵
شماره حلقه	۴۱	۴۲	۴۳	۴۴	۴۵	۴۶	۴۷	۴۸	۴۹	۵۰	۵۱	۵۲	۵۳	۵۴	۵۵	۵۶	۵۷	۵۸	۵۹	۶۰
تعداد رله در حلقه	۴	۴	۳	۵	۴	۴	۴	۵	۳	۵	۵	۶	۴	۶	۶	۷	۷	۳	۵	۵
شماره حلقه	۶۱	۶۲	۶۳	۶۴	۶۵	۶۶	۶۷	۶۸	۶۹	۷۰	۷۱	۷۲	۷۳	۷۴	۷۵	۷۶	۷۷	۷۸	۷۹	۸۰
تعداد رله در حلقه	۶	۶	۶	۵	۷	۵	۷	۵	۵	۵	۴	۶	۶	۶	۴	۶	۵	۷	۷	۵

و ماتریس NRT بصورت زیر است:

شماره رله	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
تعداد رله هم حلقه	۹۸	۱۰۳	۸۶	۸۶	۱۱۱	۹۳	۹۳	۶۹	۱۱۱	۹۳	۹۳	۹۸
شماره رله	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
تعداد رله هم حلقه	۹۸	۱۰۳	۸۶	۸۶	۱۱۱	۹۳	۹۳	۶۹	۱۱۱	۹۳	۹۳	۹۸

هر چه عدد در ماتریس NRL کوچکتر باشد، مفهوم آن اینست که حلقه دارای تعداد رله کمتری است. این بدان معنا است که در هماهنگی رله مورد نظر پس از تعداد تنظیمات کمتر باید هماهنگی اش کنترل گردد.

هر چه عدد در ماتریس NRT بزرگتر باشد، یعنی با تنظیم یک رله تعداد بیشتری رله، یا با این رله و یا بصورت شعاعی با این رله هماهنگ می‌شوند.

با توضیحات بالا، یک الگوریتم برای بدست آوردن مجموعه نقاط شکست بهینه، فقط

با توجه به تعداد هماهنگ‌ها بدست می‌آید. این الگوریتم به شرح زیر است:

۱. کوچکترین عدد ماتریس NRL را انتخاب می‌کنیم و رله‌های آن حلقه را بدست می‌آوریم و آن رله‌ای که کمترین عدد در NRT را داراست اولین نقطه شکست بهینه، فقط از دید تعداد رله‌های هماهنگ شده است. اگر چند حلقه در ماتریس NRL بودند که کمترین مقدار را داشتند، یکی از آنها را به دلخواه انتخاب می‌کنیم.

۲. تمامی حلقه‌هایی که این رله در آنها قرار دارد باز می‌شوند. (اینکار به این خاطر انجام می‌شود که وقتی یکی از اعضای حلقه‌ای را حذف کنیم کل حلقه از بین می‌رود). برای انجام اینکار، کلیه سطرهای ماتریس L_D که رله مزبور در آن قرار دارد را مساوی صفر قرار می‌دهیم.

۳. اگر ماتریس L_D صفر شد کار به اتمام رسیده است (صفر شدن ماتریس L_D یعنی اینکه دیگر حلقه‌ای در شبکه باقی نمانده است) در غیر اینصورت یک رله دیگر به عنوان دومین نقطه شکست انتخاب می‌گردد.

در مورد مثال شکل (۳)، کمترین عدد در ماتریس NRL ، عدد ۳ است که مربوط به حلقه‌های ۳ و ۹ و ۱۸ و ۴۹ و ۴۳ و ۵۸ است. یکی از این حلقه‌ها را بدلاخواه انتخاب می‌کنیم. مثلاً حلقه شماره ۳. رله‌های این حلقه را از روی ماتریس L_D بدست می‌آوریم. رله‌های این حلقه برابر رله‌های به شماره ۸ و ۱۱ و ۱۹ می‌باشد. از روی ماتریس NRT بیشترین عدد متعلق به رله شماره ۱۱ است. پس رله ۱۱ بعنوان اولین نقطه شکست معرفی می‌گردد. حال تمام حلقه‌هایی که رله ۱۱ را شامل می‌شوند، از روی ماتریس L_D بدست می‌آیند. این حلقه‌ها به شماره‌های ۲ و ۳ و ۴ و ۵ و ۶ و ۱۹ و ۲۳ و ۲۴ و ۲۷ و ۲۸ و ۳۲ و ۳۴ و ۳۸ و ۳۹ و ۵۵ و ۵۷ هستند. این سطرهای را در ماتریس L_D مساوی صفر قرار می‌دهیم. حال ماتریس NRL و NRT را از روی ماتریس جدید L_D حساب می‌کنیم.

ماتریس NRL بصورت زیر است:

شماره حلقه	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰
تعداد رله در حلقه	۴	۰	۰	۰	۰	۰	۴	۵	۳	۵	۵	۶	۴	۶	۶	۷	۷	۳	۰	۵
شماره حلقه	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱	۳۲	۳۳	۳۴	۳۵	۳۶	۳۷	۳۸	۳۹	۴۰
تعداد رله در حلقه	۶	۰	۰	۰	۷	۵	۰	۰	۵	۵	۴	۰	۶	۰	۴	۶	۵	۰	۰	۵
شماره حلقه	۴۱	۴۲	۴۳	۴۴	۴۵	۴۶	۴۷	۴۸	۴۹	۵۰	۵۱	۵۲	۵۳	۵۴	۵۵	۵۶	۵۷	۵۸	۵۹	۶۰
تعداد رله در حلقه	۴	۴	۳	۵	۴	۴	۴	۵	۳	۵	۵	۶	۴	۶	۰	۷	۰	۳	۵	۵
شماره حلقه	۶۱	۶۲	۶۳	۶۴	۶۵	۶۶	۶۷	۶۸	۶۹	۷۰	۷۱	۷۲	۷۳	۷۴	۷۵	۷۶	۷۷	۷۸	۷۹	۸۰
تعداد رله در حلقه	۶	۶	۶	۵	۷	۵	۷	۵	۵	۵	۴	۶	۶	۶	۴	۶	۵	۷	۷	۵

و ماتریس NRT بصورت زیر است:

شماره رله	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
تعداد رله هم حلقه	۶۹	۷۲	۷۴	۵۷	۸۰	۶۳	۹۳	۵۰	۷۱	۵۹	O	۹۸
شماره رله	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
تعداد رله هم حلقه	۴۹	۹۳	۹۳	۱۱۱	۶۹	۴۹	۸۱	۸۷	۷۹	۶۳	۹۰	۸۵

کمترین عدد در ماتریس NRL را بدست می‌آوریم. البته اعداد صفر را در نظر نمی‌گیریم. کمترین عدد باز عدد ۳ است که مربوط به حلقه‌های شماره ۹ و ۱۸ و ۴۳ و ۴۹ و

۵۸ است. یکی از این اعداد بدلخواه انتخاب می‌گردند. مثلاً حلقه ۹ را انتخاب می‌کنیم. رله‌های این حلقه را بدست می‌آوریم و....

با الگوریتم بدست آمده در بالا این کار را ادامه می‌دهیم. نقاط شکست رله‌های به شماره ۱۱ و ۱۴ و ۲۲ و ۲۰ و ۱۶ و ۳ و ۷ و ۶ می‌باشد.

الگوریتم کامپیوتری تئوری گراف در قسمت اصلی پروژه ارائه می‌شود. پس در این قسمت از ارائه برنامه کامپیوتری خودداری می‌گردد.

با این توضیحات تئوری گراف کامل شد. حال می‌توان روی کارهایی را که تا کنون برای بهبود پیدا کردن نقاط شکست بهینه انجام شده است، بحث کرد.

۳-۳- کارهایی که تا کنون انجام شده است

در زمینه پیدا کردن نقاط شکست فعالیتهایی به شرح زیر انجام شده است که پایه اصلی آنها همان تئوری گراف است. در بعضی مقالات روش‌هایی برای محاسبه راحتتر و سریعتر ماتریس‌های تئوری گراف داده اند [۳ و ۲]. در این مقالات راهکارهایی عنوان شده است تا تئوری گراف خیلی سریعتر به جواب برسد. در مقالات دیگری روش‌های جدیدی برای پیدا کردن تعداد وابستگی‌ها و همچنین حذف حلقه‌های شبکه عنوان شده است. یک روش دیگر استفاده از عناصری جهت باز کردن حلقه‌ها است که از عنصری مثل دیود برای نشان دادن باز شدن شبکه در جهت مخالف استفاده می‌کنند [۶]. از روش‌هایی که بر مبنای تئوری گراف نیست استفاده از توابع وابسته است. این روش بر اساس تعریف یک چندجمله‌ای از رله‌ها است [۴]. در این روش از یک نقطه کار انتخاب نقاط شکست را شروع می‌کنند و قدم به قدم کار را ادامه می‌دهند و اگر به نقاطی رسیدند که دیگر راهی برای ادامه نبود دوباره یک یا چند مرحله به عقب بر می‌گردند. در اینجا هم تنها مساله مهم توبولوژی شبکه و تعداد رله‌های هماهنگ شده است و اصلاً روی نقاط شکست بهینه کار نمی‌کنند.

با توجه به خلاصه بالا معلوم است که در راه حل‌های پیشنهادی در این زمینه، فقط به توبولوژی شبکه قدرت توجه کرده‌اند و پارامترهای تعیین کننده دیگری که بر انتخاب نقاط شکست تأثیر دارند از دید مخفی مانده است.

۳-۳- مقدمه‌ای بر هوش مصنوعی

هوش مصنوعی یک تکنولوژی نوظهور است که به صورت یک پایگاه فعال برای تحقیقات در آمده است. کاربرد عملی دانش بصیرت با محدودیت ادراکی آن می‌باشد. از هنگامیکه محققین در این زمینه شروع به کار کرده‌اند یک شاخه از هوش مصنوعی به وجود آمده است که ارائه بصیرت را با تکنیک‌های حل مسائل ترکیب نموده است. امروزه بخش موفقیت‌آمیز این شاخه تحت نام سیستمهای خبره شناخته می‌شوند. یک سیستم خبره با طراحی درست قادر است تخصص مهندسین با تجربه را در خود جمع کند و پروسه‌های اتخاذ تصمیم انسانی را به درستی تقلید نماید.

سیستم‌های نوین قدرت بوسیله اپراتورهای ماهر با بهره‌گیری از سیستم کنترل کامپیوتری که سیستم مدیریت انرژی نامیده می‌شود کنترل می‌گردد. وظیفه سیستم مدیریت انرژی تأمین عملکرد اقتصادی و حفاظتی سیستم قدرت است. بعلاوه سیستم مدیریت انرژی باید عملیات دقیقه‌ای که توسط پرسنل انجام می‌گیرد را تسهیل نماید. این سیستم عمدتاً برای کار در حالت نرمال طراحی شده است. بنابراین سیستم مدیریت انرژی در حالت اضطراری دارای راندمان کمی می‌باشد. در موقع اضطراری باید به تصمیم‌های تصحیح کننده اپراتور نیز متکی بود.

تعريف هوش مصنوعی

هوش مصنوعی شامل اصول و روش‌هایی است که بوسیله محققین زیادی که از اواخر دهه ۱۹۵۰ روی آن موضوع کار کرده‌اند، توسعه یافته است. در این زمان تعاریف زیادی برای هوش مصنوعی مطرح شد، اما هیچکدام بطور عام پذیرفته نشد. تعریفی که معمولاً پذیرفته شده است عبارتست از: هوش مصنوعی یک راه حل کامپیوتری مسائل پیچیده از طریق اعمال مراحلی مشابه پروسه استدلال انسان می‌باشد. هوش مصنوعی به صورت خیلی ساده طریقی است که قادر می‌سازد یک کامپیوتر هوشمندانه فکر کند. با مطالعه چگونگی فکر کردن بشر هنگامیکه به حل مسائل می‌پردازد، یا می‌خواهد تصمیمی را اتخاذ نماید و تقسیم بندی پروسه‌های فکر کردن به مراحل مختلف و طراحی برنامه‌های کامپیوتری که با استفاده از این مراحل به حل مسأله می‌پردازد، این عمل امکان پذیر شده است.

از آنجاییکه هوش مصنوعی ریشه در روند تفکر انسان دارد، بنابراین در این رابطه تجسس در مورد چگونگی تفکر انسان یک امر اساسی است. البته تاکنون کسی بدرستی نمی‌داند که مغز انسان چگونه کار می‌کند. هوش انسان یک عمل پیچیده است که دانشمندان فقط شروع به درک آن نموده‌اند. اما برای ما همین مقدار نیز کفايت می‌کند تا حدسه‌هائی در مورد چگونگی تفکرمان بزنیم و آن گمانها را در طراحی برنامه‌های هوش مصنوعی مورد استفاده قرار دهیم.

نتایج نهائی که همه روندهای تفکر ما به طرف آنها معطوف می‌شود، اهداف نامیده می‌شوند. بدون این اهداف، دلیلی برای فکر کردن نداریم. وقتی که سیستم هوش مصنوعی را طراحی می‌کنیم، هدف سیستم را باید همیشه در ذهن داشته و به خاطری‌سپاریم.

همه میدانیم که مغز انسان دارای حجم عظیم علوم و دانشی است که مربوط به رشته‌های غیر قابل شمارش از ایده‌ها و موضوعات می‌باشد. حیات انسان متکی به توانائی او در بکارگیری این دانش در حوادث و مسائل آینده است و همچنین متکی به یادگیری مداوم از تجربیات جدید به نحوی بوده که او را قادر ساخته تا با موقعیت‌های مشابه در آینده بروخد نماید. با تقسیم نمودن این دانش و علوم به صورت واقعیت و فرموله کردن دستجاتی از قواعد مربوط به این واقعیتها در جهت رسیدن به هدف یا اهداف می‌توان به درجه‌ای از هوشمندی رسید.

۳-۴- سیستم‌های خبره

۱-۴- مقدمه

خبرگان مردمی هستند که در مسائل بخصوصی تبحر دارند. مهارت آنها معمولاً از تجربیات وسیع و آشنایی به جزئیات مسئله‌ای که روی آن عمل می‌کند ناشی می‌شود. پزشکان مشاور و مفسران اطلاعات چاهه‌ای نفت و مهندسین متخصص که عمل تشخیص و تعییر وسائل با تکنولوژی بالا نظیر کامپیوترها و رایانه‌ها و نقائص سیستم‌های قدرت را انجام می‌دهند، نمونه‌هایی از خبرگان هستند. یک سیستم خبره، سیستمی است که :

۱. روی مسائل پیچیده و عینی که نیاز به تفسیر متخصص دارند عمل می‌کند.
۲. این مسائل را با استفاده از یک مدل کامپیوتری از استدلال انسان متخصص بررسی می‌کند و به همان نتایجی که انسان متخصص می‌تواند برسد می‌رسد. یک سیستم خبره

که بر روی یک کامپیوتر سوار می‌شود، نیاز به تجربه و دانش انسان متخصص دارد تا بتواند مانند او به حل مسائل بپردازد و هر جا که به وجود انسان متخصص نیاز باشد و تعداد متخصصین کم باشد یک مشاور بر روی کامپیوتر می‌تواند متخصص مورد نیاز را در اختیار همگان قرار دهد. یک سیستم خبره می‌تواند دانش عملی را که انسان متخصص با آن می‌تواند به حل مسائل بپردازد و در کتابها کمتر یافت می‌شود را کسب نماید. فهم اینکه واقعاً مغز انسان چگونه به حل مسائل تخصصی می‌پردازد، برای تولید موفقیت‌آمیز سیستم‌های خبره ضروری نیست. کافی است اطلاعات و دانش یک انسان متخصص را استخراج و آنها را با استفاده از یک روش ارائه کامپیوتری یکنواخت سازماندهی کنیم تا کامپیوتر بتواند عمل پردازش را برای کاربردهای مختلف انجام دهد. این بدین معنی نیست که هر مسئله‌ای که به تخصص نیاز داشته باشد می‌تواند با روش‌های سیستم‌های خبره کامپیوتری گردد. بلکه یک گروه وسیع از مسائل می‌توانند به نحو مؤثری به وسیله این روشها حل گردد.

تاریخچه سیستم‌های خبره

"اولین سیستم خبره موفق جهانی بود و در اواسط دهه ۷۰ در دانشگاه استانفورد توسعه یافت اگر "MYCIN" نبود، سیستم‌های خبره در آزمایشگاه تحقیقات هوش مصنوعی مانده بودند و گسترش به جهان خارج نداشتند. یکی از بزرگترین مسائل خیالی هوش مصنوعی این بود که بیشتر مردم شامل عده‌ای از برنامه‌سازان احساس می‌کردند که روش‌های هوش مصنوعی فقط در حوزه مسائل ساده (گاهی مسائل بازی نامیده می‌شوند) که به دسته مشخصی از قواعد و حدسهای نیازمند است، کار می‌کند. یک عده بر این باور بودند که هوش مصنوعی هرگز برای حل مسائل مشکل و حقیقی نمی‌تواند استفاده گردد. "MYCIN" همه تصورات را تغییر داد. "MYCIN" برای کمک به پزشکان متخصص جهت تشخیص بعضی از بیماری‌های باکتریایی طراحی شده است. تشخیص بیماری ضرورتاً مقایسه علائمی است که بیمار ارائه می‌دهد. با علائم یک بیماری تا وقتی که یک تطابق پیدا شود. مشکل این است که برای متخصص تشخیص سریع و معتبر برای همه بیماری‌های موجود کاری دشوار است و "MYCIN" این نیاز را به وسیله تأیید و تصدیق تشخیص برآورده ساخت. یکی دیگر از سیستم‌های خبره تجاری PROSPECTOR می‌باشد که در ۱۹۷۸ به وسیله P.Hard ، R.Reboh

ایجاد شد. *R.Duda PROSPECTOR* یک خبره در زمین‌شناسی است. این سیستم خبره احتمال پیداشدن ذخایر معدنی خاصی را در منطقه مشخصی پیش‌بینی می‌کند. از یان برنامه، برنامه‌های دیگری مشتق شده اند. شامل برنامه‌هایی که می‌توانند، کشف نفت گاز طبیعی و هلیم را پیش‌بینی کنند.

اوائل دهه ۱۹۸۰ شاهد حضور مقدماتی سیستمهای خبره‌ای بود که مشاوره مالیاتی، بیمه و حقوقی را در اختیار قرار می‌دادند. بیشتر برنامه‌سازان بر این باورند که تا آخر دهه ۱۹۹۰ یک فروشگاه بزرگ برای سیستمهای خبره شخصی که بتواند در منزل یا محل کار استفاده شوند وجود خواهد داشت. این سیستمهای زیادی از باگبانی گرفته تا تعمیر اتومبیل خبره خواهند بود. در حقیقت سیستمهای خبره می‌توانند ثابت کنند که عمومی‌ترین برنامه هستند که روی کامپیوترهای مشخص اجرا می‌شوند.

مزایای سیستمهای خبره

مطلوبیت سیستمهای خبره به این علت است که در تمام اوقات قابل دسترسی بوده و فاقد مشکلات یک انسان خبره، همچنین سیستمهای خبره را می‌توان از روی یکدیگر کپی نمود و در دسترس همگان قرار داد. انسانهای خبره با تمام مسائل عاطفی و انسانی مواجه هستند که در کار خبرگشان تأثیر می‌گذارد و به طور کلی عوامل خارجی در تصمیم‌گیری و اجرای عملیات توسط یک انسان خبره سهیم هستند ولی برای ماشین اینچنین نیست. بصیرت در یک سیستم خبره برآختی قابل نسخه‌برداری و ذخیره کردن است. در نتیجه فقدان دائمی بصیرت خبره کاملاً نادر خواهد بود. مزیت دیگر این است که خبره کامپیوتری همیشه در بالاترین سطح اجرائی قرار دارد. وقتی که یک انسان خبره احساس خستگی می‌کند قابلیت اطمینان به مشاوره او ممکن است خدشه‌دار گردد. به هر حال خبره کامپیوتری همیشه بهترین ایده‌های ممکن را در محدوده بصیرتش تولید می‌نماید.

ارائه بصیرت

یکی از چند نتیجه‌ای که در ۲۰ سال اول تحقیقات هوش مصنوعی به سختی و سریع به دست آمد، آن بود که هوش نیاز به بصیرت دارد. برای اینکه ما در ارزیابی خود اغراق نکرد، باشیم به ذکر این خواص می‌پردازیم :

۱. حجیم است.
۲. به سختی می‌توان به صورت دقیق طبقه‌بندی نمود.
۳. به طور مداوم در تغییر است.
۴. بنابراین بصیرت باید طوری ارائه شود که :
 - ❖ بتواند عمومیت را پذیرا باشد.
 - ❖ بوسیله مردمی که آن را در اختیار دارند، فهمیده شود.
 - ❖ به راحتی اصلاح شود. تا بتواند خطاهای را درست کند و تغییرات موجود در جهان را پذیرا باشد.
 - ❖ اگر چه دقیق و کامل نباشد، بتواند در محدوده وسیعی به کار گرفته شود.
 - ❖ بتواند بر مشکلاتی که ناشی از حجیم بودن اوست غلبه کند (بوسیله محدود کردن امکاناتی که باید در نظر گرفته شود).

مدلهای ارائه بصیرت، معیارهای مختلفی را که بصیرت متخصصین در یک حالت سازمان یافته بکار برد می‌شود را توضیح می‌دهند. اگر چه روش‌های مختلفی در هوش مصنوعی وجود دارد که می‌توانند به عنوان ساختار داده‌ای برای پایگاه بصیرت بکار روند. اما سیستمهای خبره از یکی از پنج مدل زیر برای بصیرت استفاده می‌نمایند:

۱. شبکه‌های وضعی (*Semantic Networks*)
۲. چهارچوبها (*Frames*)
۳. قوانین تولید (*Production Rule*)
۴. حساب محمولات (*Predicate calculus*)
۵. ترکیب موارد فوق

هر مدل ارائه بصیرت معايب و مزاياي خود را دارا می‌باشد . به طور طبیعی به روش ترکیبی نیاز است که امتیازات روش‌های دیگر را جمع‌آوری کرده و از معايب آنها بپرهیزد. میکرو کامپیوترها معمولاً از قوانین تولید استفاده می‌کنند زیرا فهم و اجرای آن ساده‌تر است. روش‌های دیگر که پیچیدگی بیشتری دارند، روی مینی کامپیوترها و کامپیوترهای بزرگ بکار گرفته می‌شوند.

هدف، انتخاب مدل برای ارائه بصیرتی می‌باشد که بتواند کاربردهای دنیای واقعی را به خوبی تقلید نماید. برای هر مسئله بخصوص متخصص قبلاً یک ساختار تعریف شده یا مدل منطقی از حوزه عمل (حتی اگر صریح نباشد) را دارد.

انتخاب مدل ارائه بصیرت که به دنیای متخصص نزدیک باشد، عمل وارد کردن بصیرت در کامپیوتر را راحت‌تر می‌سازد. علاوه بر آن اشکال‌یابی و آزمایش سیستم خبره بسیار آسانتر خواهد بود. بنابراین انتخاب مدل ارائه بصیرت در هنگام ساخت سیستم خبره بسیار مهم می‌باشد.

اجزاء یک سیستم خبره

سیستم‌های خبره از معماری‌های مختلف سیستم استفاده می‌کنند. زیرا برای یک مورد خاص ممکن است یک معماری کاراتر از سایر معماری‌ها باشد. هم‌اکنون تحقیقات وسیعی برای بررسی جنبه‌های مختلف معماری‌های سیستم‌های خبره در حال جریان است و بحث در آن زمینه هنوز ادامه دارد. علی‌رغم اختلافات مهم بین معماری‌ها، اغلب آنها اجزاء مشترک دارند. جزئیات هر جزء در بخش‌های زیر توضیح داده شده است.

(Knowledge Base) پایگاه بصیرت

پایگاه بصیرت شامل ارتباطات بین داده‌های مسأله می‌باشد. قدرت یک سیستم خبره به پایگاه بصیرت آن بستگی دارد. سیستم‌های خبره ای موفق هستند که در بصیرت غنی باشند، حتی اگر قابلیت حل مسأله ضعیفی داشته باشند. این بصیرت به طور وسیع آزمایشی، غیر مطمئن و قابل داوری نسبی می‌باشد. در حال کلی بصیرت، به صورت حقایق و قوانین ذخیره می‌شود. اما برای ذخیره اطلاعات (ارائه بصیرت از حالت‌های بخصوص استفاده می‌شود). پروسه کسب بصیرت در یک حوزه بخصوص و وارد کردن آن در پایگاه بصیرت به مهندسی بصیرت شهرت دارد. مهم‌ترین بخش در ساخت یک سیستم خبره کسب بصیرت می‌باشد.

موتور استنتاج

سیستم‌های خبره باید بتوانند در موقعیت‌های مختلف کار کنند. قابلیت پاسخگویی به شرایط متغیر بستگی به استنتاج بصیرت جدید از بصیرت فعلی دارد. پروسه جستجو برای یافتن

یک بصیرت و همچنین استنتاج بصیرت‌های مختلف و جدید یک جزء اساسی سیستم‌های خبره است. یکی از بزرگترین مشکلات در عملکرد سیستم‌های خبره این است که ترکیب اجزاء منفرد می‌تواند منجر به ایجاد اجزاء ترکیب شده بسیاری گردد. اگر یک مجموعه بزرگ از عناصر وجود داشته باشد، تعداد و حالات ممکن بصورت نجومی بالا خواهد رفت. این مسئله به انفجار ترکیبی مرسوم است. برای فائق آمدن بر این مسئله اغلب سیستم‌های خبره از بصیرت کامپایل شده (یک بصیرت سطح بالا که از طریق تجربه به صورت ضمنی تولید می‌شود) به جای کار کردن روی عناصر اولیه بصیرت استفاده می‌کنند.

موتور استنتاج باید راهی جستجو، حل کردن مسئله و استدلال از طریق پایگاه بصیرت برای انجام قیاس منطقی، استنتاج و یا اثبات یک راه حل داشته باشد. موتور استنتاج باید بداند که برای حل مسئله از کجا شروع کند و تناظرها را پیش آمده را چگونه حل کند. روش یکه کامپیوتر برای حل مسئله و استدلال انتخاب می‌کند، بستگی به نوع مسئله و مدل ارائه بصیرت دارد. در واقع موتور استنتاج سعی دارد که روش انسان متخصص را در مواجهه با مسائل شبیه‌سازی کند. روش‌های جستجو و استراتژیهای مختلف استدلال در فصل قبل ذکر شد است.

وابل کاربر (User Interface)

وظیفه وابل کاربر این است که اطلاعات کاربر را به فرم قابل قبول برای سیستم خبره تبدیل کند، یا آنکه اطلاعات فراهم شده از طریق سیستم خبره را برای کاربر قابل فهم سازد. به صورت ایده‌آل وابل کاربر شامل یک سیستم پردازش زبان طبیعی است که اطلاعات را می‌پذیرد و یا بر می‌گرداند به شکلی که یک متخصص انسانی این عمل را انجام می‌دهد. در حالی که امروزه هیچ سیستمی وجود ندارد که بتواند تمام قابلیت‌های زبان طبیعی را داشته باشد. اما سیستم‌های کارآمدی وجود دارند که می‌توانند تسهیلاتی در استفاده از یک زبان محدود و یک زیر مجموعه از زبان طبیعی را فراهم نمایند. وابل کاربر برای سیستم‌های خبره اغلب برای تشخیص حالتی که کاربر کار می‌کند و سطح تخصص کاربر و طبیعت کار بکار بردۀ می‌شود.

وابل‌های کاربر شامل یک ترمینال، پنجره‌های گرافیکی چندحالته، پنجره‌های کاراکتری می‌باشد.

واصل کاربر در سه موقعیت عمل می‌کنند:

۱. کاربر به عنوان مشتری عمل می‌کند . در این مورد کاربر جوابی برای مسئله‌اش می‌خواهد.

۲. کاربر به عنوان آموزشگر برای افزایش قابلیت‌های سیستم خبره عمل می‌کند.

۳. کاربر به عنوان شاگرد سیستم خبره عمل می‌کند.

واصل بین کاربر و برنامه به طور کامل بستگی به نوع عمل و سخت‌افزار استفاده شده دارد.

طراحی سیستمهای خبره (Expert System Design)

ساختن یک سیستم خبره با کد کردن یک برنامه تفاوت زیادی دارد . سیستم خبره یک میدان جدید برای محاسبه است . ساخت یک سیستم خبره در سه مرحله انجام می‌شود که در ذیل به شرح آنها می‌پردازیم.

شناسایی و تعریف مسئله

اولین مرحله شناسایی مسئله است. این مرحله به مراحل زیر قابل تقسیم است:

- ❖ مرور مسئله : یک لیست از مسائل تهیه کنید.
- ❖ کاندید کردن یک انتخاب : لیست بالا را کاهش دهید.
- ❖ دسترس پذیر بودن متخصص: یک متخصص مناسب باید در دسترس باشد.

ساختن نمونه

به محض آنکه حوزه مسئله تعیین شد، عمل بعدی ساخت یک نمونه است که بخش کوچکی از سیستم نهایی است.

مراحل آن عبارتند از:

- ❖ کسب بصیرت اولیه
- ❖ نزدیک شدن به مسئله اساسی
- ❖ مدل مشاوره کلی

- ❖ انتخاب چهارچوب استنتاج
- ❖ انتخاب ارائه بصیرت
- ❖ انتخاب ابزار
- ❖ اجرای نمونه
- ❖ آزمایش نمونه
- ❖ نمایش نمونه
- ❖ اصلاحات و تجدید نظر پروژه

ساختن سیستم خبره

به محض آنکه نمونه ساخته شده و به مورد اجرا در آمد ، طراح به طراحی کل سیستم خبره می پردازد . مراحل اجرای این قسمت به شرح زیر است :

- ❖ اصلاح نمونه
- ❖ ساخت چهارچوب سیستم
- ❖ کسب بصیرت اصلی
- ❖ ساخت نرم افزار فرعی
- ❖ مجتمع کردن داخلی
- ❖ اصلاح داخلی

۳-۴-۵- آینده سیستم های خبره

گرایش های اخیر در ساخت سیستم های خبره بستگی به فاکتورهای اجتماعی و تکنیکی پیدا کرده است. ساخت سیستم های خبره همواره تحت ریزبینی هایی انجام شده است که مهارت های انسان متخصص، همراه با مهارت های متخصص کامپیوتر را با هم به خدمت گرفته است. برای آنکه مشخص شود یک سیستم ساخته شده متخصص است، باید میزان کارایی آن مشخص باشد. همه بصیرت ها نمی توانند با قوانین تولید نمایش داده شوند.

بعلاوه متخصص ممکن است از اطلاعات محلی، بصیرت ساختار یا عملکرد یا ارتباطات ریاضی در حل مسئله استفاده کند. هیچ کدام از انواع اطلاعاتی که گفته شد، نمی تواند

به سادگی با قوانین تولید نمایش داده شود. مطمئناً با داشتن یک مجموعه ارتباطات غنی در پایگاه بصیرت سیستم خبره بهتری خواهیم داشت.

هدف نهایی طرح کسب بصیرت ایجاد قابلیتی است که متخصص بتواند بصیرت خود را دقیقاً در سیستم کامپیوتری وارد نماید. اغلب نتایج عملی در آینده نزدیک احتمالاً از کاربرد سیستم های خبره فعلی در مواجهه با مسائل دنیای واقعی بدست خواهد آمد. آینده، زمان استفاده از سیستم های خبره به طور وسیع خواهد بود.

۳-۴-۳- کاربرد سیستمهای خبره قدرت

امروزه هدف مشترک در مدیریت سیستمهای قدرت این است که نیروی الکتریکی ارزانتر، بی خطرتر و با اطمینان بالاتر تولید شود. این امر به کمک سیستمهای کنترل اتوماتیک مانند ردیابی همزمان باز متغیر در سیستم، بهینه‌سازی تولید برای به حداقل رساندن هزینه نیروگاهها و همکاری عملیات مرکز مستقل کنترل انجام پذیر می‌باشد.

کنترل و عملکرد سیستمهای نوین قدرت بوسیله مهندسین متخصص و با تجربه تحت حمایت یک سیستم کنترل کامپیوتری به نام سیستم مدیریت انرژی ("EMS") می‌باشد. این سیستم در مرکز سیستم کنترل اصلی که ساختمان آن سلسله مراتبی می‌باشد واقع شده است. از ترمینالهای خارج از مرکز، شبکه مخابراتی و سطوح مختلف سیستم‌های پردازش کامپیوتری استفاده می‌کند. هدف سیستم مدیریت انرژی تأمین عملکرد اقتصادی و حفاظتی سیستم قدرت و تسهیل انجام عملیات دقیقه‌به‌دقیقه‌ای است که توسط پرسنل مرکز کنترل انجام می‌شود. سیستم مدیریت انرژی عمدتاً برای کار در حالت نرمال طراحی شده است و عملیاتی از قبیل تخمین حالت، تجزیه و تحلیل حفاظتی و جریان قدرت بهینه را که برای عملکرد مطمئن سیستم می‌باشد، محاسبه می‌کند. همچنین برای تأمین عملکرد اقتصادی سیستم قدرت عملیاتی نظیر کنترل تولید اتوماتیک، پخش بار اقتصادی، بکارگیری واحدهای تولید و پیش‌بینی بار توسط سیستم‌های مدیریت انرژی انجام می‌شود. تقریباً تمامی اتفاقات در حالت نرمال بوسیله کامپیوتر کنترل شده و نیروی انسانی (اپراتور) فقط برای انجام تعداد محدودی از فرمانهای دستی در کار کامپیوتر مداخله می‌نماید. در حالتهای اضطراری مانند زمانی که یک رویداد غیر قابل پیش‌بینی رخ می‌هد و یا یکی از اجزای اصلی سیستم از کار می‌افتد، وضعیت و عملکرد سیستم مدیریت انرژی کاملاً متفاوت می‌باشد. در چنین وضعیتهای اضطراری سیستم

مدیریت انرژی مانند یک سیستم جمع‌آوری اطلاعات و گزارشگر عمل می‌نماید و تمام برنامه‌های قابل و گرانقیمت که در حالت نرمال بسیار مفید بودند، دیگر کاربرد چندانی ندارند. همچنین در مواردی که به طور ناگهانی یکی از اجزای انتقال نیرو از مدار خارج می‌شود، این اپراتور است که باید بداند چه اتفاقی رخ داده است و تصمیم بگیرد که برای رفع اشکال چه باید کرد. بنابراین نرم‌افزارهای معمولی برای حالت اضطراری کاربرد چندانی ندارند. باید در این موقع از برنامه‌های تصمیم گیرنده یا با هوش استفاده کرد.

از عهده رخدادهای اضطراری برآمدن به نام "پروسه تشخیص و تصمیم گیری" شناخته شده است. اینگونه پروسه‌ها "بدساختار" گفته می‌شوند. به طور کلی برای اینگونه پروسه‌های "بدساختار" راه حل عمومی وجود ندارد. بیشتر به تجربه و مهارت اپراتور در تشخیص و تصمیم گیری بستگی دارد. قابلیت اپراتور در اینگونه وضعیت‌ها بستگی به تجربه و ابتکار او دارد، که آیا در گذشته چنین وضعیتی را دیده است و عملکردی شبیه به این داشته است یا خیر؟ در ضمن اپراتور باید بتواند وضعیت کامپیوتری موجود را با رویدادهای ثبت شده قبلی مقایسه کند تا به نتیجه مطلوب برسد. چون تمام عملیات اپراتور در وضعیت‌های اضطراری و پروسه تشخیص و تصمیم گیری باید بسیار سریع و "بهنگام" باشد، بنابراین اپراتورها معمولاً تحت تأثیر عوامل خارجی و بالنتیجه فشار فکری قرار می‌گیرند. این فشار باعث پائین آمدن قدرت تفکر و درک آنها می‌شود. به اینگونه فشارهای فکری در هنگام تصمیم گیری "سد ادراکی" گفته می‌شود. همچنانکه پیچیدگی عملیات و اجرای تصمیمات در سیستمهای قدرت افزایش می‌یابد، این سد ادراکی نیز بیشتر احساس شده و قدرت اپراتور برای تصمیم گیری کاهش می‌یابد. تاکنون سعی و تلاش کافی و به حد نیاز برای کم کردن این سد ادراکی انجام نشده است و نیاز مبرمی برای تحقیقات در این زمینه می‌باشد.

هوش مصنوعی بخصوص سیستمهای خبره اجازه کاربرد کامپیوتر را در تکنیکهای ابتکاری برای پروسه‌های بد ساختار را به ما می‌دهد. محرک اصلی برای استفاده از سیستمهای خبره در قدرت احتیاج به از بین بردن سد ادراکی اپراتور در وضعیت‌های اضطراری می‌باشد.

یکی از راههایی که به اپراتور کمک می‌کند سد ادراکی را از میان بردارد اینست که در اینگونه موارد استثنائی (که اپراتور باید بداند چه اتفاقاتی در شبکه رخ داده است) سیستم خبره می‌تواند اپراتور را با نشان دادن سناریوهای متعدد از سیستم و توضیح هر سناریو راهنمایی کرده و موقعیت سیستم را به طور دقیق برای او شرح دهد. بنابراین اپراتور می‌تواند سناریوهایی را که معقول و باورکردنی هستند برای اجرای عملیات انتخاب کند. این در حالتی است که

بدون سیستم خبره بعضی از این سناریوها می‌توانست از فکر اپراتور دور بماند و نتواند در آن مورد خاص تصمیم اتخاذ نماید. از آنجا که عملیات بر روی یک سیستم قدرت از یک سلسله اعمال یکسان و روزمره تشکیل شده که به طور عادی توسط اپراتور انجام می‌شود، به کمک سیستم خبره می‌توان بعضی از اعمال عادی و یکنواخت را به کامپیوتر سپرد تا اینکه اپراتور وقت بیشتری برای کارهای مهمتر داشته باشد. بنابراین می‌توان دید که سیستم خبره در صنعت برق می‌تواند نقش مؤثر و مفیدی را ایفا نماید و برای ادامه تحقیقات در این زمینه کاندید معتبری می‌باشد.

هم‌اکنون تکنولوژی سیستم خبره نظر بسیاری از افراد جامعه مهندسین قدرت، محققین دانشگاهها، ارگانهای تحقیقاتی و مهندسین و اپراتورهای کمپانی‌های بزرگ قدرت را به خود جلب نموده است. تا امروز تقریباً تمامی سیستمهای خبره حالت نمونه برای به نمایش درآوردن قابلیت آن در سیستم‌های قدرت و یا تحقیقات در این زمینه را دارند و راه درازی برای به نتیجه رساندن این تکنولوژی نوین در پیش است.

کاربرد سیستم خبره در قدرت می‌تواند یکی از اصول موفق برای بالا بردن سطح مدیریت سیستمهای انرژی باشد و نمونه‌هایی از آن که در مقالات متعددی گزارش شده در زیر نام برده شده است:

- » طراحی پخش بار (*Load Flow Planning*)
- » پیش‌بینی بار (*Load Forecasting*)
- » بکارگیری واحدهای تولید (*Unit Commitment*)
- » پروسس کردن علائم اخباری (*Alarm Processing*)
- » تعیین حفاظت سیستم (*Security Assessment*)
- » کنترل ولتاژ و توان راکتیو (*Reactive Power and Voltage Control*)
- » استقرار مجدد سیستم (*System Restoration*)
- » تشخیص خطا (*Fault Diagnosis*)
- » پایداری گذرا (*Transient Stability*)
- » پخش بار (*Real Power Flow*)
- » کنترل ولتاژ (*Voltage Control*)
- » قطع بار (*Load Shedding*)
- » تشخیص عیوب دستگاهها (*Failure Diagnosis of Equipment*)

- پخش بار اقتصادی (*Economic Dispatch*)
- تعیین به هم پیوستگی شبکه (*Topology Determination*)
- رله‌های حفاظتی دیجیتال و باهوش (Intelligent Digital Relay Protection)

۳- استفاده از سیستم خبره در پیدا کردن نقاط شکست

۱- دلیل استفاده از سیستم خبره در بدست آوردن نقاط شکست

نظر باینکه تا بحال روش یا روشهایی که برای یافتن نقاط شروع هماهنگی (نقاط شکست) بکار رفته است مبتنی بر تئوری گراف است، در اینجا به ناکامی این روش و لزوم جایگزینی یک روشی مبتنی بر مبنای سیستم خبره در پیدا کردن نقطه شکست می‌پردازیم. برای این منظور اشکالاتی که در سیستم تئوری گراف موجود است را عنوان می‌کنیم و نشان می‌دهیم که سیستم خبره برای این منظور راهکار مفیدی است. بدین منظور به دسته‌بندی اشکالات این روش پرداخته شده است اشکالات مطرح شده در این قسمت، نسبت به تئوری گراف بدین نیستند. یعنی با این فرض که تئوری گراف در هر صورتی بتواند در پیدا کردن نقاط شکست راهکاری پیشنهاد کند، از آن استفاده شده است، ولی در مواردی تئوری گراف دیگر کمکی در این زمینه نمی‌تواند بکند. این مسئله بدان معنی است که تئوری گراف در بعضی موارد کلا دچار مشکل می‌شود و از ارائه دادن مجموعه نقاط شکست قادر می‌ماند و یا مجموعه نقاط شکستی که ارائه می‌کند، به هیچ صورت رله‌های شبکه را هماهنگ نمی‌کند. پس اشکالاتی که عنوان شده است اشکالات اساسی هستند که تا کنون تئوری گراف راه حلی برای آنها ارائه نداده است. این اشکالات به شرح زیر هستند.

۱-۱-۳- در نظر نگرفتن پارامترهای سیستم قدرت در تئوری گراف

تا کنون دیدیم که تمام روش‌هایی که برای پیدا کردن نقاط شکست به کار رفته است به توپولوژی مدار قدرت ربط دارد و هیچکدام به عناصر موقعیت رله‌ها کار ندارد. ملحوظ نکردن این پارامترها به وضوح جواب‌هایی می‌دهد که در بعضی اوقات کاملاً غیر قابل قبول است.

مثلاً در نظر گرفته شود که در یک شبکه، در نقطه‌ای یک فیوز یا یک عنصر سریع بکار رفته است. می‌دانیم که فیوز نمی‌تواند پشتیبان رله باشد. یعنی این نقطه، باید نقطه شکست باشد. این گفته بدان معنی است که فیوز پشتیبان هیچ رله‌ای نباشد و رله‌های دیگر باید پشتیبان فیوز یا عنصر باشند. یا در نظر گرفته شود که در نقطه‌ای قدرت اتصال کوتاه برای شینه‌ای بسیار بالا است. این بدان مفهوم است که تنظیمات رله‌های این شینه بیشترین *TSM* را دارا هستند. در حقیقت تنظیمات رله‌های این شینه باید زودتر از رله‌های دیگر تعیین می‌شود تا اطمینان حاصل شود که رله‌های پشتیبان کار خود را بخوبی انجام می‌دهند.

پارامترهایی مثل پارامترهای توضیح داده شده در بالا بسیارند که در تئوری گراف این پارامترها در نظر گرفته نشده‌اند. تعدادی از این پارامترها از قبیل نوع بارها و حفاظت پایلوت و ... است.

۱-۲-۳- برابر بودن اعداد در ماتریس NRL و NRT :

در بعضی شبکه‌ها، پس از محاسبه تعداد وابستگی‌های هر رله در روش معمولی و تئوری گراف، چند عدد مساوی داریم. یعنی چند رله هستند که تعداد وابستگی‌های یکسانی دارند. حال اینکه کدام رله انتخاب شود مساله‌ای است که با تئوری گراف نمی‌توان جواب قطعی به آن داد.

در بسیاری از شبکه‌ها، پس از محاسبه ماتریس NRL ، به چند عدد مساوی برخورد می‌کنیم. یعنی چندین حلقه وجود دارند که تعداد رله‌های در هر جهت آنها با هم مساوی است. حال اینکه حلقه کدام رله به عنوان حلقه شروع کننده معروفی شود سؤالی است که تئوری گراف پاسخی برای آن ندارد. مثلاً در نظر گرفته شود که یک حلقه‌ای انتخاب شود. این حلقه بدلخواه انتخاب می‌گردد. حال اگر کار ادامه داده شود، دوباره ممکن است ماتریس NRL جدید، درایه‌های مساوی باشد. یک حلقه دوباره به دلخواه انتخاب می‌شود و این کار ادامه داده می‌شود. با توضیحات بالا معلوم است که تعداد حالت‌هایی که بدلخواه حلقه‌ای انتخاب می‌گردد بسیار زیاد است و جواب ممکن است بسیار نامعقول باشد. مشکل گفته شده در بالا برای ماتریس NRL نیز برقرار است. یعنی در بسیاری از شبکه‌ها، پس از محاسبه ماتریس NRL ، به چند عدد مساوی برخورد می‌کنیم. یعنی رله وجود دارند که در یک حلقه هستند و تعداد رله‌های پشتیبان آنها در همان حلقه و حلقه‌های مجاور برابر است. حال آنکه کدام رله انتخاب شود، جای سؤال است.

دو مشکل گفته شده در بالا در حقیقت دارای تاثیرات متفاوتی هستند. یعنی مشکل گفته شده در ابتدا (برای ماتریس NRL) بسیار حادتر است. این بدان دلیل است که نهایت کار، انتخاب رله است. هر چه در مرحله نزدیکتر به هدف انتخابهای دلخواه باشد، بهتر از این است که در مراحل دور از هدف انتخابهای دلخواه باشد. ماتریس NRL فقط حلقه با تعداد رله کمتر را پیدا می‌کند. این انتخاب، غیر مستقیم به انتخاب رله بستگی دارد و در مرحله دورتر از انتخاب رله که در NRT است قرار دارد. پس خطای NRL باعث دور شدن بسیار از جواب اصلی می‌شود. از آنجا که تعداد حلقه‌های با تعداد رله برابر زیاد است، مشخص است که خطای بسیاری در این قسمت رخ خواهد داد. پس باید راه حلی ارائه کرد که راه حل انتخاب حلقه از روی NRL به انتخاب رله وصل شود. بطوريکه واسطه‌ای مثل انتخاب حلقه با تعداد رله کمتر نتواند تغییرات و اشتباهات زیادی در انتخاب رله ایجاد کند.

۱-۳-۳- کار نکردن تئوری گراف به عنوان یک بخش مستقل

یکی از راه حلهایی که ممکن است در نظر آید این است که مبنای کار هنوز تئوری گراف باشد. لیکن خروجی‌های تئوری گراف توسط سیستم خبره پردازش گردد. بدین صورت که در ابتدا تئوری گراف تمامی مجموعه نقاط شکست شبکه را بدون توجه به تعداد رله‌های وابسته و تعداد رله در هر حلقه محاسبه بکند. بعد از اینکار، سیستم خبره وارد شود و بهترین نقطه شکست را تعیین کند.

این فکر از آنجا ناشی شد که اشکالاتی که به تئوری گراف وارد شده است، بسیار خوش‌بینانه است. در این قسمت صحت این گفته معلوم می‌شود. در واقع راه حل پیشنهادی، فرض کرده است که تئوری گراف بتواند تمامی حالت‌هایی که حلقه‌های شبکه در دو سوی ساعتگرد و پاد ساعتگرد باز می‌شوند را به کاربر نشان دهد و یا تمام جوابها را در فایلی بریزد. این بدان معنی است که کار تئوری گراف بدون بهینه کردن جواب انجام می‌شود. پس در حقیقت پایه اصلی، همان تئوری گراف قرار گیرد.

پس از این مرحله یک سیستم خبره روی این فایل آنالیزهای لازم را انجام دهد و بهترین نقطه شکست را به عنوان خروجی نهایی بدهد. یعنی این سیستم خبره به هر کدام از پارامترهای گفته شده امتیازی بدهد و روی هر مجموعه رله‌ای که از تئوری گراف بدست می‌آید این امتیازها را اعمال کند. حال از بین این مجموعه‌ها، آن مجموعه‌ای که بالاترین امتیاز را دارد، انتخاب شود و بعنوان مجموعه نقاط شکست بهینه معرفی گردد.

برای تشریح راه حل بالا این روش بر روی شبکه شکل (۳) انجام شد.

یعنی با استفاده از تئوری گراف تمامی مجموعه رله‌هایی که می‌توانند حلقه‌های شبکه را در دو جهت ساعتگرد و پاد ساعتگرد باز کنند محاسبه شد. با اجرای این راه حل بر شبکه شکل (۳)، نتیجه در حدود ۴۰۰۰۰۰ مجموعه نقطه شکست شد. مفهوم این جواب این است که ۴۰۰۰۰ مجموعه رله متشکل از این ۲۴ رله می‌باشد که با حذف شاخه‌های گراف شبکه در جهت رله‌های آنها، حلقه‌های شبکه باز می‌شوند. با توجه به اینکه تعداد شینه‌ها بسیار کم بوده است، مشخص است که این راه حل برای شبکه‌های واقعی که بمراتب دارای تعداد خطوط و شینه‌های بیشتری است عملاً مجموعه بسیار بسیار وسیعی را بعنوان نقاط شکست شبکه پیشنهاد خواهد کرد که تحلیل و آنالیز آنها توسط سیستم خبره غیر ممکن است. پس تئوری

گراف بعنوان بخشی از سیستم خبره و در قوانین خبره بکار برده می‌شود تا در پیدا کردن هر نقطه شکست، بهترین را انتخاب کند.

گفته‌های بالا بدین مفهوم است که تئوری گراف نمی‌تواند تمام مجموعه نقاط شکست را بدهد. بلکه باید با الگوریتمی که در جلوتر پیشنهاد داده می‌شود، در دل سیستم خبره بنشیند تا در هر مرحله تعیین نقاط شکست به سیستم خبره کمک کند تا نقطه شکست را بدست آورده و بعد به سراغ نقطه شکست دیگر برود، تا جایی که کلیه حلقه‌های شبکه در دو جهت باز شوند.

۱-۳-۴- یکی بودن امتیاز ماتریس‌های NRL و NRT

در تئوری گراف پس از محاسبه ماتریس تعداد رله‌های هر حلقه (NRL) کمترین عدد این ماتریس مورد ملاحظه قرار می‌گیرند. سپس رله‌های آن حلقه تعیین می‌گردند و بزرگترین عدد در ماتریس تعداد رله‌های وابسته به هر رله NRT پیدا می‌شوند. حال اگر دو عدد در ماتریس NRL با هم مساوی بودند حلقه‌ای در نظر گرفته می‌شود که رله‌های آن در NRL اعداد بالاتری دارند. یعنی امتیاز اعداد در ماتریس NRL بسیار بالاتر از امتیاز اعداد در ماتریس NRT است. در صورتی که می‌تواند این امتیازها بنا به شرائط مختلف تغییر کند. این مسئله را می‌توان در سیستم خبره با وزن دادن به این ماتریس‌ها حل کرد.

فرض کنید یک حلقه دارای سه رله و یک حلقه دارای چهار رله است. ولی حلقه‌ای که دارای چهار رله است دارای رله‌ای است که تعداد پشتیبانهای آن خیلی بیشتر از تعداد پشتیبانهای حلقه‌ای است که دارای سه رله است. این بدان معنی است که در حلقه با چهار رله، رله‌ای وجود دارد که اگر تنظیمات آن معین گردد، تنظیمات تعداد زیادی از رله‌های شبکه معلوم می‌گردد. ولی حلقه با سه رله دارای این مشخصه نیست. یعنی رله‌ای ندارد که با تنظیم این رله، این تعداد رله از سیستم هماهنگی را با مشخص شدن تنظیماتشان حذف کند. ولی در سیستم تئوری گراف نقطه رله‌های حلقه با سه رله بحث می‌شود.

با گفته‌های بالا، نتیجه جالبی عاید می‌شود آن اینست که امتیاز ماتریس NRL در تعیین نقطه شکست بسیار بالاتر از ماتریس NRT می‌باشد. این فرض باعث مشکلات توضیح داده شده در بالا می‌گردد. برای حل این مشکل باید کاری کرد که دو مشکل از راه برداشته شود. اول آنکه به نوعی ماتریس NRL را به رله‌ها مربوط کنیم ثانیا ضریب دو ماتریس NRL و NRT را متفاوت بگیریم تا به جواب معقولتری برسیم.

۱-۳-۵- عوض کردن نقاط شکست در صورت نرسیدن به جواب

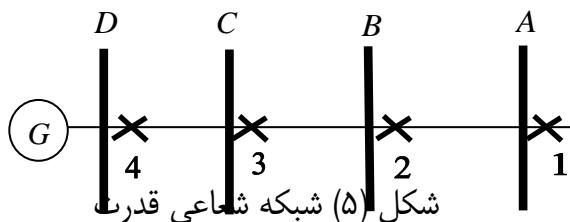
در نظر گرفته شود برای تعیین مجموعه نقاط شکست، با استفاده از تئوری گراف به یک جواب رسیدیم. ولی در آخر پس از هماهنگی رله‌ها، این هماهنگی مطلوب نبود. مثلاً دو رله در جایی با هم هماهنگ نشدند و یا در نقطه‌ای که سطح اتصال کوتاه بالا بود، تنظیمات طوری بودند که رله دیر عمل می‌کرد. با این عیبها باید بتوان مجموعه نقاط شکست جدیدی بدست آورد. این مجموعه اولاً باید دارای مشخصات نقطه شکست باشد. ولی در این مرحله باید مشکل ایجاد شده را بیشتر مورد اهمیت قرار دهند. یعنی امتیاز آن پارامتر را در تعیین مجموعه نقاط شکست افزایش دهیم. در تئوری گراف برای این مسئله هیچکونه تمھیداتی نگذاشته است. ولی با در نظر گرفتن امتیاز برای هر پارامتر می‌توان این مشکل را مرتفع کرد. بدین صورت که برای هر رله بخاطر هر پارامتر امتیاز در نظر گرفته می‌شود و مجموعه نقاط شکست تعیین می‌گردد. اگر پس از هماهنگی رله‌ها به ناهمانگی و یا هماهنگی بد برخورد کردیم، نوع ناهمانگی و یا هماهنگی بد و پارامتری که باعث شده است این ناهمانگی و یا هماهنگی بد ایجاد شود را تشخیص داده و دوباره به سیستم خبره برگشته و ضریب این عوض می‌گردد تا در تعیین نقاط شکست این پارامتر بهتر و با اثر بیشتر در نظر گرفته شود.

۳-۲- قوانین سیستم خبره

با توجه به صحبت های گفته شده ، ورود یک سیستم خبره برای پیدا کردن نقاط شکست اجباری است. حال مساله اساسی، تعیین قوانین سیستم خبره است. این قوانین که در این قسمت مطرح می‌شوند، هیچکدام صدرصد تعیین کننده نقطه شکست نیستند. بلکه می‌توان با ضریبی بگویند که رله مربوطه می‌تواند نقطه شکست باشد یا نه و یا با چه ضریبی کاندیدای نقطه شکست بودن است. قوانینی که در اینجا مطرح می‌شوند محدودند، ولی راه را برای ورود هر قانون دیگری باز گذاشته‌اند. یعنی با شناختن قوانین جدید می‌توان آنها را به ذیل قوانین زیر اضافه کرد تا بتوان هر چه بهتر و سریعتر به جواب مورد نظر دست پیدا کرد. در ذیل این قوانین آورده شده است. برای هر کدام از قوانین، اول توضیح قانون آمده است، سپس توضیح داده شده است که اساساً چرا این قانون وارد سیستم خبره شده است. ثالثاً طرز اعمال آن به برنامه کامپیوتری آمده است، یعنی چگونه از روی گراف و اطلاعات شبکه، قانون را بصورت ریاضی برای ماتریسهای تئوری گراف بیان کرد.

۱-۲-۳ - دور بودن و نزدیک بودن به منبع

اولین قانونی که در این قسمت بحث می‌شود، دور بودن و نزدیک بودن به منبع است. قبل از شروع این مبحث، به توضیح مثال ساده‌ای در شبکه‌های ساعی می‌پردازیم. یک شبکه ساعی مانند شکل زیر در نظر گرفته شود.

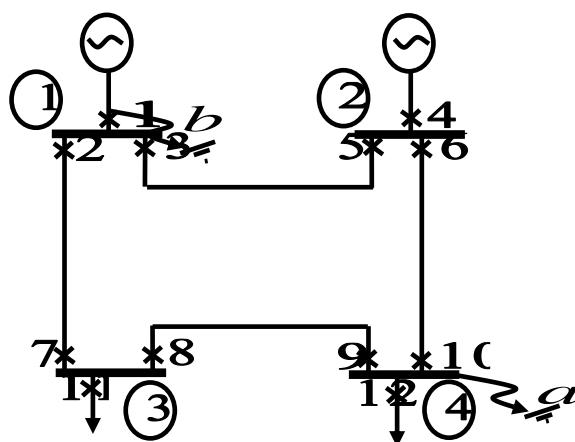


می‌دانیم برای تنظیم خوب، باید اول رله شماره یک را به عنوان نقطه شکست انتخاب کنیم. یعنی نقطه شروع تنظیمات از رله یک است. با توجه به مفهوم نقاط شکست، می‌توان گفت رله یک نقطه شکست است. البته در اینجا مفهوم نقطه شکست از دید باز شدن حلقه‌های شبکه نیست، بلکه از دید رله‌ای است که اولین تنظیمات را دارا است و تنظیمات دیگر بر روی آن قرار می‌گیرند.

اگر بخواهیم را حل بالا را به عنوان یکی از قوانین خبره معرفی کنیم به صورت زیر عمل می‌کنیم:

هر چه رله از منبع دورتر باشد، احتمال نقطه شکست بودن آنجا بیشتر است.
می‌توان گفت که هر چه رله از منبع دورتر باشد، پس سطح اتصال کوتاه پائینتر است.
پس اگر رله‌های آن شینه‌ها تنظیم گردند، رله‌های شینه‌های دیگر راحت‌تر تنظیم می‌گردد تا به رله‌های نزدیک منبع بررسیم.

ولی یک نکته مهم در این قسمت وجود دارد. در حقیقت نمی‌توان برای تمام رله‌هایی که دور از منبع هستند این امتیاز را قائل شد. برای تشریح این مفهوم به شرح یک مثال پرداخته می‌شود. شکل شماره (۵) را در نظر گرفته شود.

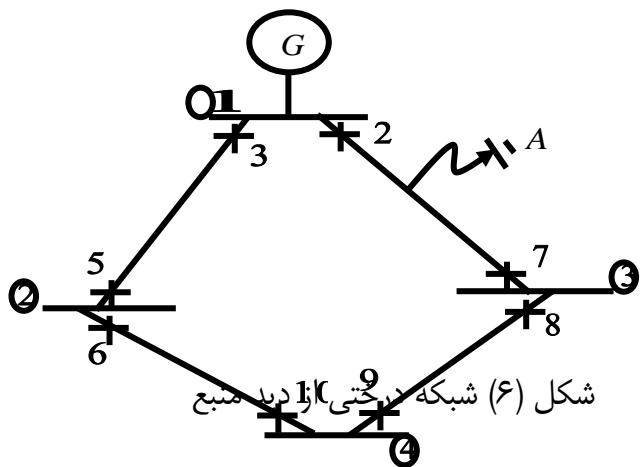


شکل (۵) شبکه قدرت نمونه برای امتیاز دوری از منبع

در این شکل رله‌های شماره ۲ و ۳ امتیاز نزدیکی به منبع را می‌گیرند ولی رله شماره ۱ امتیاز نزدیکی به منبع را نمی‌گیرد. این مسئله از روی شکل (۴) و با مقایسه خطا در نقاط a و b مشخص است. چون رله ۱ وقتی وارد سیستم هماهنگی می‌شود که در b خطا رخ دهد و این خطا در حقیقت منبع را از شبکه جدا می‌کند و منبع روی عملکرد رله ۱ تأثیر نمی‌گذارد.

برای توضیح بیشتر فرض کنید خطا در یک نقطه از شبکه اتفاق بیافتد. بروز خطا در یک نقطه باعث می‌شود که تمام جریانهای شبکه، از منابع مختلف به آنجا برسند. حال فرض کنید دارای یک منبع است. این فرض غلط نمی‌باشد. چون با استفاده از جمع آثار می‌توان شبکه را اینگونه فرض کرد که هر منبع را به تنها یکبار حساب کنیم و منابع دیگر را در هر مرحله نادیده گرفت. سپس بین آنها می‌نیمم امتیازها را گرفت.

اگر شبکه دارای یک منبع باشد، خطا در هر نقطه باعث می‌شود که جریانها از منبع به آن نقطه سرازیر شوند. برای ساده‌تر شدن مفهوم آن، در نظر بگیرید که منبع در بالای صفحه قرار دارد و خطوط ارتباطی شینها، منبع را به شینهای مرتبه اول (شینهایی که با یک خط به منبع وصل هستند) و مرتبه دوم وصل می‌کنند. برای شکل (۵) می‌توان با در نظر گرفتن منبع شین ۱ و حذف منبع شین ۲، شکلی مطابق شکل زیر داشت.



در این شکل شینهای ۲ و ۳ شینهای مرتبه اول و شین ۴ شین مرتبه ۲ است. آنچه مشخص است اینست که رله‌های ۲ و ۳ امتیاز دوری از منبع را می‌گیرند ولی رله‌های ۷ و ۵ این امتیاز را نمی‌گیرند. چون اگر خطابی رخ دهد، جریانها طوری نیستند که رله ۷ کاملاً آنها را ببینند.

از جهتی در نظر بگیریم خطی از شین ۲ به شین ۳ وصل باشد. اگر خطابی در وسط شین ۱ و ۳ مثلاً A رخ دهد، از مسیر شین ۱ و ۲ و ۳ جریانی نمی‌گذرد که رله ۷ آنرا می‌بندد. پس رله ۷ نیز امتیاز دوری از منبع می‌گیرد. پس دقت شود که دور بودن از منبع در جهت رله از هر مسیری را باید حساب کنیم و به این فاصله‌ها امتیاز بدھیم، نه دور بودن شین از منبع. پس برای اینکه بتوانیم این قانون را وارد برنامه کامپیوتری کنیم از الگوریتم زیر استفاده می‌کنیم:

- (۱) در ابتدا در ورودی برنامه منابع معلوم می‌گرددند.
- (۲) یکی از منابع را در نظر می‌گیریم.
- (۳) توبولوژی شبکه به صورت درخت درآورده می‌شود که در سطر اول آن منبع باشد.
- (۴) فاصله هر رله را در جهت خودش با این منبع بدست می‌آوریم.
- (۵) از بین فاصله‌های حساب شده، فاصله می‌نیم را حساب می‌کنیم.
- (۶) امتیاز رله برای این پارامتر متناسب با فاصله است. (این گفته بدان معنی است که هر چه رله دورتر باشد، امتیازش برای نقطه شکست شدن بیشتر است).
- (۷) منبع دیگری را انتخاب می‌کنیم.
- (۸) به مرحله ۳ بر می‌گردیم.
- (۹) در آخر بین امتیازها کمترین عدد را در نظر می‌گیریم.

۳-۲-۳- سطح اتصال کوتاه

قانون دومی که در اینجا بحث میشود، از آنجا آمده است که اگر بین دو شین، قدرت اتصال کوتاه یکی بالاتر از دیگری باشد، اول رله مربوط به شین با اتصال کوتاه پایین‌تر تنظیم می‌گردد. به طور مثال شبکه شکل (۵) را در نظر بگیرید. در این شبکه با توجه به گفته‌های بخش (۱-۲-۳) اول رله مربوط به رله (۱) تنظیم می‌گردد. سپس رله (۲) و ... این کار بدین دلیل است که اگر رله (۱) را روی پایین‌ترین TSM بگذاریم، رله‌های دیگر را می‌توان با های بالاتر تنظیم کرد. چون جریان اتصال کوتاه در رله‌های دیگر بیشتر از جریان اتصال کوتاه در رله شماره (۱) است. از همین ایده می‌توان برای شبکه‌های دیگر نیز استفاده کرد. صورت قانون را می‌توان به صورت زیر بیان کرد:

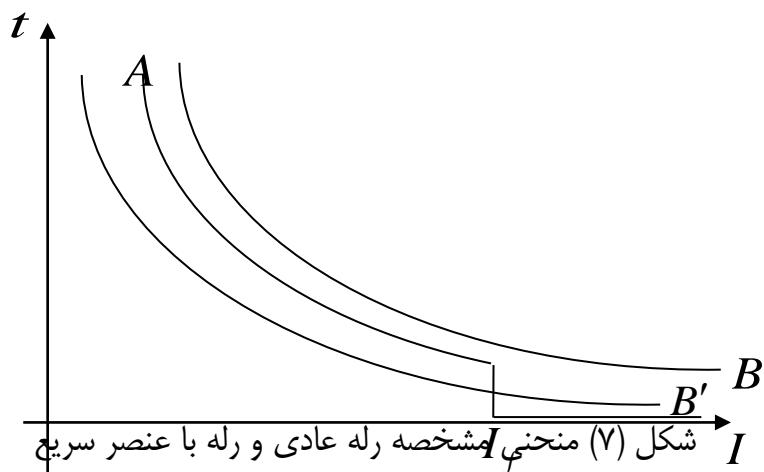
هر چه سطح اتصال کوتاه پایین‌تر باشد، احتمال بودن نقطه شکست بیشتر است.
ایده استفاده شده در بالا، برای ورود به برنامه کامپیوتری دارای الگوریتم ساده‌ای است.
اگر خطا را جلوی هر رله قرار دهیم و جریان اتصال کوتاه را حساب کنیم، مشخص می‌شود که هر رله چه مقدار جریان را در لحظه خطا می‌بیند. سپس این جریانها را با هم مقایسه می‌کنیم.
هر چه جریان اتصال کوتاه بالاتر باشد، امتیاز رله برای نقطه شکست بودن پایین‌تر است.
یعنی TSM ‌های مربوط به رله‌ای که دارای جریان اتصال کوتاه بالاتر است، بالاتر است.
پس TSM کمتر که به احتمال زیادتری نقطه شکست است مربوط با جریان اتصال کوتاه کمتر است. پس الگوریتم کامپیوتری زیر را داریم.

- خطا را جلوی هر رله می‌گذاریم
- جریان خطا را حساب می‌کنیم
- اینکار را برای تمامی رله‌ها انجام می‌دهیم
- هر چه جریان اتصال کوتاه بالاتر باشد، امتیاز رله برای نقطه شکست بودن پایین‌تر است. پس با مقایسه جریانهای اتصال کوتاه، امتیازی به هر رله می‌دهیم.

۳-۲-۳- عنصر سریع

قانون سومی که در این قسمت بحث می‌شود، مربوط به یکی از قانونهای هماهنگی رله‌ها می‌باشد. یکی از قوانینی که در هماهنگی رله‌ها استفاده می‌شود اینست که رله‌ای که سریعتر عمل می‌کند، نمی‌تواند پشتیبان رله‌ای که کندر عمل می‌کند باشد. مثلاً در سیستم شعاعی اولین رله TSM کمتر را می‌گیرد. حال رله‌های دیگر با TSM بالاتر می‌توانند پشتیبان این رله که TSM آن کمتر است بشوند. این توضیح این نکته را به ما نشان می‌دهد که اگر عنصر سریع در مکانی قرار بگیرد، کمتر رله‌ای می‌تواند رله اصلی آن باشد. یعنی رله با عنصر سریع نمی‌تواند پشتیبان رله‌های دیگر که معمولاً کندر است، بشود. منظور از عنصر سریع در این قسمت می‌تواند فیوز و یا رله با عنصر سریع باشد.

از دید مشخصه رله‌ها فرض کنید منحنی A در شکل ۷ مشخصه رله با عنصر سریع و منحنی B در شکل ۷ مشخصه رله بدون عنصر سریع باشد.



عنصر سریع در یک مکان یعنی اینکه احتمال بودن نقطه شکست در آنجا بالاست. این دلیل به خاطر آنست که بالا بودن سرعت عملکرد رله باعث این می‌شود این رله نتواند پشتیبان رله‌های دیگر که احتمالاً سرعت پایین تری دارند باشد. پس خود این نقطه شرایط خوبی برای نقطه شکست بودن دارد.

طرز ورود به برنامه اینگونه است که در ورودی برنامه، عناصر سریع با کدی مشخص می‌شوند که در حقیقت امتیازی برای اینست که رله مورد نظر نقطه شکست باشد.

۳-۲-۴- تعداد رله‌های هم حلقه

قانونی که در این قسمت گفته می‌شود، همان قانونی است که در تئوری گراف از آن استفاده می‌شود. برای تشریح این قانون در نظر بگیرید که رله A در حلقه‌های L_1 و L_2 قرار دارد اگر رله A به عنوان نقطه شکست معرفی گردد، هر دو حلقه L_1 و L_2 را در جهت خودش باز می‌کند. پس تمام رله‌های حلقه L_1 و L_2 که در جهت این رله هستند را می‌توان با این رله هماهنگ کرد. پس در حالت کلی اگر رله A در چند حلقه L_1 و L_2 و ... L_n مشترک باشد، با انتخاب رله A به عنوان نقطه شکست می‌توان تمام حلقه‌ها را در جهت آن رله باز کرد و تمامی رله‌های دیگر تنظیماتان معلوم می‌گردد.

از جهتی هر چه تعداد نقطه شکست بالاتر باشد، مشکلات بیشتری در هماهنگی رخ می‌دهد. این مسئله از آنجا ناشی می‌شود که پس از اینکه رله‌های شبکه هماهنگ شد، هماهنگی رله‌های نقطه شکست با رله‌های پشتیبانشان باید چک شود. در بعضی اوقات ممکن است در بین این هماهنگی به ناهماهنگی یا هماهنگی بد برخورد کنیم. پس هر چه تعداد این چک کردنها کمتر باشد، برای نرسیدن به ناهماهنگی یا هماهنگی بد احتمال کمتری وجود دارد. پس بهترین جواب آن جوابی است که تعداد نقاط شکست در مجموعه نقاط شکست آن کمتر باشد.

با دو موضوع مطرح شده در بالا، مشخص است که هر چه رله‌ای که انتخاب می‌شود دارای تعداد رله هم حلقه بیشتری باشد، در هر دفعه تعداد بیشتری از رله‌ها از سیستم هماهنگی خارج می‌شوند و شبکه باقیمانده دارای تعداد رله کمتری می‌شود. پس با سرعت بیشتری به انتهای کار که باز شدن حلقه‌ها در دو سر است می‌رسیم و تعداد نقاط شکست کمتری بدست می‌آوریم.

با گفته‌های بالا قانون این قسمت بصورت زیر در می‌آید:

هر چه تعداد رله هائی را که لازم است با این رله هماهنگ شوند بیشتر باشد نقطه شکست بهتری است.

به طور خلاصه دلیل این امر این است که اگر این رله تنظیم شود، تعداد بیشتری رله هماهنگ می‌شود و از سیستم هماهنگی خارج می‌شوند. برای ورود این قانون به برنامه اینگونه عمل می‌کنیم که با استفاده از تئوری گراف و با کمک ماتریس‌های تئوری گراف پیدا می‌کنیم که هر رله با چند رله دیگر هم حلقه است. سپس

به نسبت تعداد وابستگی به هر رله امتیاز می‌دهیم. یعنی هر چه رله تعداد رله هم‌حلقه بیشتر داشته باشد امتیاز بالاتری می‌دهیم. پس الگوریتم کامپیووتری زیر را برای این قسمت بکار می‌بریم.

- یک رله را انتخاب می‌کنیم.
- تعداد رله‌های هم‌حلقه این رله را بدست می‌آوریم.
- امتیازی متناسب با این تعداد به رله‌ها می‌دهیم.
- اینکار را برای تمام رله‌ها انجام می‌دهیم.

۳-۲-۵- حلقه با تعداد رله کمتر

قانون پنجمی که در این قسمت به آن می‌پردازیم، قانونی است که در اول این فصل بکی از مشکلات تئوری گراف به شمار رفت. در این قسمت با تصحیح این قانون، استفاده خوبی از این قانون خواهد شد.

در اول این فصل گفتیم که در انتخاب نقطه شکست در تئوری گراف، کوچکترین عدد در ماتریس NRT را به دست می‌آوریم. سپس رله‌های آن را انتخاب می‌کنیم. از بین آنها رله‌ای که دارای بالاترین تعداد رله هم‌حلقه است را به عنوان نقطه شکست قبول می‌کنیم. در همان قسمت عنوان شد که کم بودن تعداد رله‌ها در یک حلقه، چون بطور غیر مستقیم با نقاط شکست رابطه دارد، در انتخاب نقطه شکست دارای خطای زیادی می‌شود مخصوصاً اگر چند عدد در ماتریس NRT با هم مساوی باشند. حال می‌خواهیم قانونی را وضع کنیم که بطور مستقیم به رله وصل شود و اول انتخاب حلقه نباشد.

قانونی که در تئوری گراف برای انتخاب عدد کوچکتر در ماتریس NRT بود، به صورت

زیر بود:

نقطه شکست از رله‌های حلقه‌ای انتخاب می‌شود که تعداد رله در آن کمتر است.
حال با کمی تغییر قانون بالا را به صورت زیر می‌گوئیم:
اگر یک رله در حلقه‌ای قرار بگیرد که تعداد رله‌های آن حلقه کم باشد، برای نقطه شکست مطلوب‌تر است.

پس در حقیقت می‌توان گفت که امتیازی را می‌توان به رله‌های هر حلقه بطور یکسان داد. آن امتیاز متناسب با عکس تعداد رله‌های موجود در آن حلقه است. ولی فرض کنیم چند حلقه وجود دارند که رله A را شامل می‌شوند. برای امتیاز دادن به رله A چند امتیاز برای رله A وجود دارد. حال باید دید از روی این اعداد کدام را باید انتخاب کرد.

برای توضیح این قسمت فرض می‌کنیم که رله A در دو حلقه قرار داشته باشد که یک حلقه دارای ۴ رله و یک حلقه دارای ۵ رله باشد. اگر رله A نقطه شکست بشود هماهنگی باید هم برای حلقه‌ای که دارای ۴ رله است و هم برای حلقه‌ای که دارای ۵ رله است هماهنگی چک شود. پس اگر عدد بزرگ‌تر در نظر گرفته شود منظور را از وضعیت چک کردن برای ۴ رله می‌پوشاند. پس با این توضیحات می‌فهمیم که عدد بزرگ‌تر را در حلقه‌ها برای هر رله در نظر می‌گیریم.

برای اعمال این قانون به برنامه اینگونه عمل می‌کنیم که با استفاده از تئوری گراف تعداد رله‌های هر حلقه را بدست می‌آوریم. سپس تعداد رله در هر حلقه را بدست می‌آوریم. بعد برای هر رله ببینیم در کدام حلقه است. حلقه‌ای که بالاترین تعداد رله در بین این حلقه‌ها دارد را بدست می‌آوریم. به نسبت عکس تعداد رله مذبور به رله امتیاز می‌دهیم. الگوریتم کامپیوتری زیر دقیقاً این الگوریتم را بیان می‌کند:

- ماتریس NRT را حساب می‌کنیم.
- یک رله انتخاب می‌کنیم.
- تمام حلقه‌هایی که این رله در آنها وجود دارد انتخاب می‌کنیم.
- تعداد رله‌های هر حلقه را بدست آورده و کمترین را انتخاب می‌کنیم.
- متناسب با عکس این بالاترین عدد، امتیازی به رله می‌دهیم.
- اینکار را برای تمامی رله‌ها انجام می‌دهیم.

۳-۲-ع- داشتن یا نداشتن حفاظت پایلوت

قانون ششمی که در این قسمت بحث می‌شود، برای استفاده بهینه از تجهیزات موجود در شبکه می‌باشد. در توضیح این روش، باید مقدمه‌ای را عنوان کرد. در بسیاری از شبکه‌ها تا قبل از اینکه رله‌های شبکه از راه تئوری و درستی هماهنگ شوند معمولاً یک سری رله در شبکه قرار می‌دهند که پایه فکری درستی پشت آنها نیست. بعد از مدتی به فکر تنظیم تئوری و دقیق رله‌ها می‌افتد. ولی در شبکه تعدادی رله از انواع مختلف وجود دارد که می‌توان در هماهنگی رله‌ها از آنها استفاده بهینه کرد. مثلاً حفاظتهای پایلوت را می‌توان در هماهنگی رله‌ها استفاده کرد. زیرا می‌دانیم در بعضی موارد، پس از تعیین مجموعه نقاط شکست و شروع هماهنگی به حالتی می‌رسیم که آخرین رله هماهنگ نمی‌شود و یا یک ناحیه بدون رله اصلی یا رله پشتیبان می‌ماند. در چنین حالتی یک حفاظت پایلوت می‌تواند مشکل را حل کند. یعنی بودن حفاظت پایلوت در شبکه می‌تواند هماهنگی رله‌ها را با درصد بیشتری تضمین کند. مثلاً فرض گرفته شود رله A نقطه شکست انتخاب شود. پس از هماهنگی باید که تنظیمات رله A با رله جلوئی چک شود. فرض کنید در این چک کردن به ناهمانگی یا هماهنگی بد برسيم. می‌توانيم با گذاشتن حفاظت پایلوت اين ناهمانگي يا هماهنگي بد را از بين بيريم. پس اگر سعی شود حفاظتهای پایلوت نزديك شينه دور رله‌ای باشد که نقطه شکست است، اين حفاظتها می‌توانند در جاهایی که ناهمانگی بین رله‌های تنظیمی در آخر کار قرار دارد، به برنامه هماهنگی کمک کند.

برای اعمال این قانون به برنامه کامپیوتری اینگونه عمل می‌کنیم که رله‌هایی که در شينه دور آنها حفاظت پایلوت موجود است امتیاز بیشتری نسبت به رله‌هایی که در شينه دور آنها حفاظت پایلوت نیست دارند.

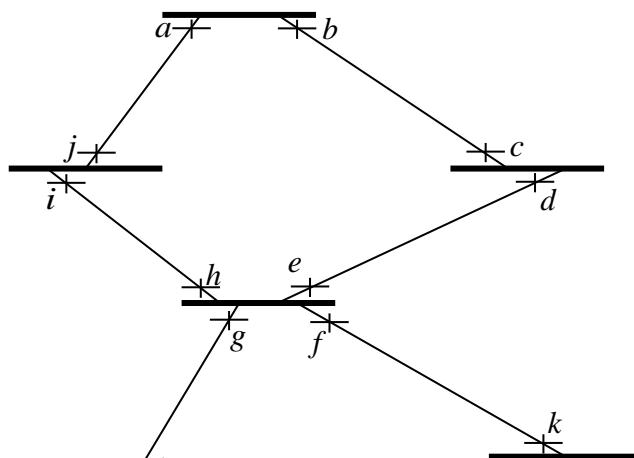
پس با استفاده از ماتریسهای گراف شينه دور هر رله را بدهست می‌آوريم و سپس چک می‌کنیم که در آنجا حفاظت پایلوت وجود دارد یا نه و امتیاز اين قانون را به رله می‌دهیم. البته می‌توان از طرف دیگر وارد شد یعنی ابتدا حفاظت پایلوت را پیدا کرد و سپس با استفاده از ماتریسهای گراف تمامی رله‌هایی که اين شينه، شينه دور آنها است امتیاز حفاظت پایلوت را بگيرند. پس الگوريتم زير را برای برنامه کامپیوتری به کار می‌بريم:

- حفاظتهای پایلوت را پیدا می‌کنیم.

- رله‌هایی که این شینه، شینه دور آنها حساب می‌شود را با استفاده از ماتیس‌های گراف پیدا می‌کنیم.
- به این رله‌ها امتیاز می‌دهیم.
- این کار را برای تمام رله‌ها انجام می‌دهیم.

۳-۲-۷- تعداد خطوط رسیده به یک شینه

هفتمین قانونی که در این قسمت بحث می‌کنیم، تعداد خطوط رسیده به یک شینه است شبکه‌ای مانند زیر در نظر گرفته شود در این شبکه شینه ۱ دارای ۴ خط است شینه ۲ و ۳ دارای ۲ خط است.



شکل (۸) قسمتی از یک شبکه ~~مانند~~

در نظر گرفته شود که تعداد رله‌های اصلی رله ز را می‌خواهیم پیدا کنیم. رله b رله اصلی رله z است. حال فرض کنیم که تعداد پشتیبانهای رله i را می‌خواهیم. رله‌های g و f رله‌های اصلی رله b است یا مثلا فرض کنید رله‌های پشتیبان رله z را می‌خواهیم. رله‌های l و h پشتیبانهای این رله هستند. در مثال آخر دیدیم که رله z دارای تعداد رله پشتیبان زیادی است. اگر دقت کنیم، این تعداد رله پشتیبان بالا مساوی تعداد خطوط رسیده به شینه دور رله در جهت عکس عملکرد رله است.

با توجه به توضیحات بالا در نظر بگیریم که رله در شینه دور عکس جهت عملکرد خودش دارای تعدادی خط باشد. هر کدام از این خطوط دارای رله‌ای مربوط به این شینه هستند. رله‌ایی که در روی همین خطوط ولی مربوط به شینه دور رله هستند، همگی پشتیبانهای رله مذبور هستند.

از جهتی می‌دانیم که هر چه تعداد رله پشتیبان رله‌ای بیشتر باشد، با هماهنگ کردن این رله تعداد بیشتری از رله‌ها از سیستم هماهنگی خارج می‌شوند.
با توضیحات بالا می‌توان قانون زیر را برقرار کرد:

هر چه تعداد خطوط رسیده به شینه‌های دور رله‌ای در عکس جهت عملکردش بیشتر باشد، امتیاز بالاتری برای نقطه شکست بودن دارد.

بطور خلاصه دلیل قانون بالا را می‌توان تعداد رله پشتیبان بالاتر عنوان کرد.

برای ورود این قانون به برنامه سیستم خبره، اول باسه‌های دور هر رله را بدست می‌آوریم (البته بدون در نظر گرفتن باس دور رله در جهت عملکرد رله). سپس مجموع خطوط رسیده به این باسها را حساب می‌کنیم. متناظر با این تعداد خطوط امتیازی به رله می‌دهیم.

الگوریتم کامپیوتروی این قانون را به صورت زیر می‌توان بیان کرد:

- یک رله را انتخاب می‌کنیم.
- باسه‌های دور رله را بجز باس دور رله در جهت عکس عملکرد رله پیدا می‌کنیم.
- تعداد خطوط رسیده بع این باسها را حساب می‌کنیم.
- متناسب با این تعداد خطوط امتیازی به این رله می‌دهیم.
- اینکار را برای تمامی رله‌ها انجام می‌دهیم.

۱-۲-۳- نوع بارها

هشتمین قانونی که در این قسمت بحث می‌شود مربوط به انواع بارهای موجود در شبکه قدرت است. برای تشریح این قانون یک کوره القایی بزرگی که وظیفه آن ذوب فلزات است در نظر بگیرید. اگر کوره در حال کار باشد، مقدار زیادی آهن مذاب در آن قرار دارد. واضح است که قطع برق کوره حتی به مدت زمان کوتاهی ضایعه خطرناکی برای کوره حساب می‌شود و باعث خسارات زیادی می‌گردد. پس در حالت کلی می‌توان بارها از نظر قطع و وصل به چند دسته تقسیم کرد:

۱. بارهایی که نباید قطع شوند.
۲. بارهایی که قطع آنها زیان زیادی ندارد.
۳. بارهایی که قطع آنها مهم نیست.

حال در نظر بگیرید رله‌ای به عنوان نقطه شکست انتخاب شد. پس اولین تنظیم را به آن می‌دهیم. یعنی سرعت آنرا بالا در نظر می‌گیریم.

حال فرض کنید در نقطه‌ای خطأ رخ دهد. رله‌های با سرعت بالاتر عمل می‌کنند. حتی اگر خطأ به سرعت برداسته شود، ممکن است که این رله‌ها عمل کنند و سیستم را از قسمت خطأ جدا کند.

با توضیحات بالا منطقی است که سعی کنیم وقتی باری نباید قطع شود، پس نقطه شکست از آن بار دور باشد تا رله آن بار اولین رله‌ای نباشد که باید عمل کند. یعنی اگر در فاصله‌ای نسبتاً دور خطأ رخ دهد، رله این بارها نباید به سرعت عمل کند.

پس قانون این قسمت را می‌توانیم به صورت زیر بیان کنیم:

هر چه رله از بارهای غیر قابل قطع دورتر باشد، احتمال نقطه شکست بودن بالاتر است.

خلاصه دلیل قانون بالا را می‌توان به صورت زیر بیان کرد: هر چه رله به بارهای غیر قابل قطع نزدیکتر باشد باید دیرتر عمل کند، چون رله‌های نقطه شکست زودتر عمل می‌کنند. برای اعمال این پارامتر به برنامه سیستم خبره، اول بارهای با مشخصه بدون قطع معلوم می‌گرددن. حال اگر بتوانیم توپولوژی شبکه را به صورت درختی درآوریم که در سطر اول آن بارهای با مشخصه بدون قطع باشد، نقاط انتهائی درخت احتمال زیادتری برای نقطه شکست بودن دارد.

الگوریتم کامپیووتری زیر می‌تواند برای ورود این پارامتر به سیستم خبره بکار رود:

- بارهایی که دارای مشخصه غیر قابل قطع هستند را پیدا می‌کنیم.
- شبکه را به صورت درختی در می‌آوریم.
- فاصله هر رله را تا این بار پیدا می‌کنیم.
- متناسب با این فاصله امتیازی به این رله می‌دهیم.
- اینکار را برای تمام بارهای غیر قابل قطع تکرار می‌کنیم.
- بین این امتیازها کمترین عدد را انتخاب می‌کنیم.

۳-۳- استنتاج

قوانينی که در قسمتهای قبل گفته شد به هر رله امتیازی برای هر پارامتر می‌دهد. پس از اینکه تمامی امتیازها برای رله‌های مختلف بدست آمد، جدولی بدست می‌آید که در آن به هر رله ۸ نوع امتیاز داده شده است. این امتیازها همان امتیازی هستند که به إزای ۸ پارامتر به رله‌ها داده شده است.

حال باید به صورتی از روی این ۸ پارامتر و یا از روی این ۸ امتیاز تصمیم نهایی را گرفت. اساس که در این قسمت استفاده شده است اینست که اول همه امتیازها را نرمالیزه به سطح یک بکنیم. یعنی در هر قسمت و به إزای هر پارامتر امتیازی که وجود دارد را برابر بالاترین عدد تقسیم می‌کنیم. نتیجه اینست که به إزای هر پارامتر یکسری امتیاز بدست می‌آید که همگی بین اعداد صفر تا یک هستند.

پس از اینکه امتیازها نرمالیزه شد، حال قسمت بسیار مهمی از استنتاج باید انجام گیرد و آن وزن دادن به هر نوع امتیاز رله است. این وزنها بصورت زیر تعیین می‌شوند:
در ابتدا باید دید که چه نوع هماهنگی را داریم. یعنی با توجه به نوع هماهنگی و اینکه در هماهنگی رله‌ها چه پارامتری مهمتر است و همچنین تجربیات در شبکه‌های قبلی این وزنها انتخاب می‌شوند. مثلاً در نظر بگیرید در شبکه‌ای برای تنظیم رله‌ایش تأکید شده است که به بارهای غیر قابل قطع دقت فراوان شود. نتیجه این می‌شود که وزن این پارامتر را در نتیجه‌گیری نهایی زیاد می‌کنیم و یا مثلاً در نظر بگیرید که در شبکه‌ای حفاظتهاي پايilot موجود است و از هماهنگ کننده رله‌ها خواسته شده است که تا جای ممکن از این حفاظتها استفاده کنند. نتیجه این می‌شود که امتیاز این پارامتر را در نتیجه‌گیری نهایی زیاد می‌کنیم و همینطور است برای وزن دادن به امتیازهای دیگر.

پس از اینکه وزنهای امتیازات مشخص شد، امتیازهای نرمالیزه شده را در وزن مربوطه ضرب کرده و حاصلجمع امتیازها در وزنها را برای هر رله حساب می‌کنیم. حال بالاترین امتیاز را در نظر می‌گیریم. در حقیقت بالاترین امتیاز به ما نشان می‌دهد که رله مذبور برای نقطه شکست بودن نسبت به رله‌های دیگر از امتیاز بیشتری برخوردار است. این رله اولین نقطه شکست می‌شود.

پس از تعیین اولین نقطه شکست، برای بدست آوردن نقطه شکست بعدی باید تغییراتی در برنامه بدهیم. این تغییرات به شرح زیرند:

در ابتدا ماتریس‌های مربوط به (حلقه با تعداد رله کمتر) و (تعداد رله هم حلقه) و (تعداد خطوط رسیده به یک شینه) را تصحیح می‌کنیم. این کار بدین دلیل است که با کم شدن یک رله از شبکه فقط این سه پارامتر دستخوش تغییر می‌گردد. سپس دوباره کار امتیازدهی را ادامه می‌دهیم و نقاط شکست بعدی را حساب می‌کنیم. این کار تا وقتی ادامه دارد که حلقه‌ای در شبکه باقی نماند. معیار این اتمام، صفر شدن ماتریس L_D تئوری گراف است.

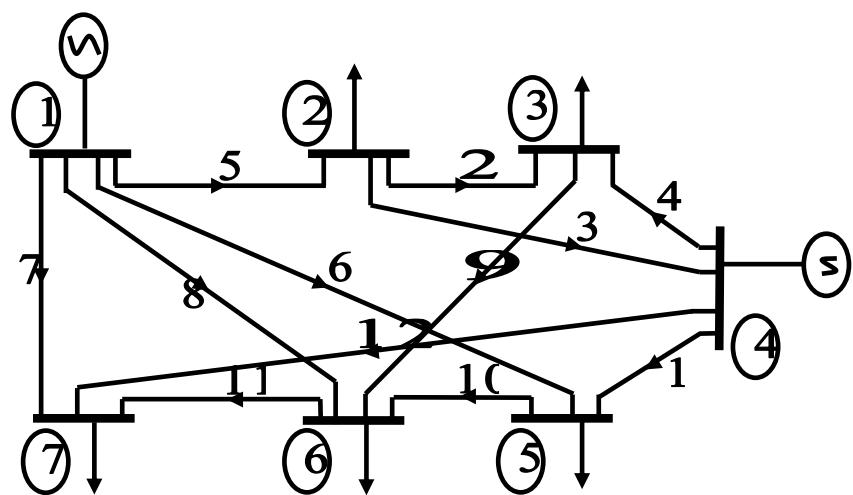
الگوریتم کامپیوتری زیر را می‌توان برای بدست آوردن نقاط شکست بیان کرد:

- اطلاعات شبکه را دریافت می‌کنیم.
- امتیاز هر پارامتر را برای تمامی رله‌ها حساب می‌کنیم.
- امتیاز هر پارامتر را برای تمامی رله‌ها بر بالاترین امتیاز رله برای این پارامتر تقسیم می‌کنیم (نرمالیزه کردن امتیازات).
- با توجه به نوع شبکه و درجه اهمیت پارامترها، وزنهایی برای هر پارامتر تعیین می‌کنیم.
- حاصلجمع امتیاز در وزن را برای هر رله حساب می‌کنیم.
- بالاترین امتیاز نقطه شکست خواهد بود.
- شبکه را در جهت این رله باز می‌کنیم.
- در ماتریس L_D گراف، تمامی حلقه‌هایی که این رله در آنها قرار دارد را حذف می‌کنیم.
- تمامی کارهای بالا را برای شبکه جدید و ماتریس L_D جدید تکرار می‌کنیم.
- اینکار را تا زمانی ادامه می‌دهیم که حلقه‌ای در شبکه باقی نماند (صفر شدن ماتریس L_D گراف).

۳-۴- آزمایش برنامه بر چند شبکه نمونه

(۱) شبکه اول مورد آزمایش:

شبکه اول مورد آزمایش بصورت شکل (۸) است.



شکل (۸) شبکه قدرت مورد آزمایش

برای آنکه بتوان برنامه نوشته شده را بر روی شبکه بالا پیاده کرد، باید تمامی اطلاعات شبکه اعم از داشتن عنصر سریع و یا سطوح اتصال کوتاه و ... را در روی شبکه معلوم کنیم. ولی از آنجا که مشخص کردن آنها در روی شبکه شلوغ می‌شود، اطلاعات را در جدول زیر قرار داده‌ایم. این اطلاعات مورد نیاز برای برنامه کامپیوتری در جدول (۱) آمده است.

پیدا کردن نقاط شکست

۶۳

جدول (۱) : اطلاعات سیستم قدرت

شماره شانه	گره ابتدا	گره انتهای	دلتین عصر سریع راه وصل به شینه ابتدا	دلتین عصر سریع راه وصل به شینه انتهای	سطح اتصال کوتاه جلوی راه ابتدا	سطح اتصال کوتاه جلوی راه انتهای	سطح اتصال کوتاه جلوی راه ابتدا	وصل بودن شینه ابتدایه منبع	وصل بودن شینه انتهایه منبع	وصل بودن شینه از نظر بار	مهم بودن شینه از نظر بار
1	4	5	0	0	4000	6000	1	1	0	0	1
2	2	3	0	1	2000	6000	0	1	0	0	1
3	2	4	0	0	4000	7000	1	1	1	0	0
4	4	3	0	0	4000	6000	0	1	0	0	1
5	1	2	0	0	1000	5000	1	0	1	0	0
6	1	5	0	1	5000	5000	0	0	0	0	1
7	1	7	0	1	8000	8000	1	0	0	0	0
8	1	6	1	0	4000	2000	0	0	0	0	0
9	3	6	1	0	4000	5000	0	0	0	1	0
10	5	6	1	0	5000	5000	0	0	0	1	0
11	6	7	0	0	5000	4000	0	0	0	0	0
12	4	7	0	0	7000	5000	0	1	0	0	0

پیدا کردن نقاط شکست

۶۴

توضیح اعداد بالا بصورت ذیل است:

با ورود این اطلاعات به برنامه گراف، امتیازهای رله‌های شبکه برای پارامترهای متفاوت حساب می‌شود. این امتیازها در جدول (۲) آمده است. پس از وزن دادن به هر پارامتر حاصل نیز در جدول (۲) آمده است.

جدول (۲) : امتیاز رله‌های شبکه

شماره رله	1	2	3	4	5	6	7	8	9	10	11	12
امتیاز پارامتر اول	3	3	1	3	0	2	2	2	4	2	0	3
امتیاز پارامتر دوم	4	2	4	4	1	5	8	4	4	5	5	7
امتیاز پارامتر سوم	0	0	0	0	0	0	0	1	1	1	0	0
امتیاز پارامتر چهارم	98	103	86	86	111	93	93	69	111	93	93	98
امتیاز پارامتر پنجم	0.16	0.16	0.2	0.2	0.16	0.16	0.2	0.2	0.1	0.16	0.16	0.16
امتیاز پارامتر ششم	1	0	1	0	1	0	1	0	0	0	0	0
امتیاز پارامتر هفتم	11	10	10	11	12	12	12	12	10	11	12	11
امتیاز پارامتر هشتم	4	5	5	4	5	5	5	5	4	4	4	4
نتیجه	8.77	8.36	8.8	8.3	8.21	8.63	10	9.08	9.08	9.3	7.97	8.14

شماره رله	13	14	15	16	17	18	19	20	21	22	23	24
امتیاز پارامتر اول	0	2	1	2	3	2	2	0	0	0	2	0
امتیاز پارامتر دوم	6	6	7	6	5	5	8	2	5	5	4	5
امتیاز پارامتر سوم	0	0	1	0	0	1	1	0	0	0	0	0
امتیاز پارامتر چهارم	98	103	86	86	111	93	93	69	111	93	93	98
امتیاز پارامتر پنجم	3	4	4	4	3	4	4	4	3	4	4	3
امتیاز پارامتر ششم	0.16	0.2	0.16	0.16	0.16	0.1	0.16	0.16	0.16	0.2	0.16	0.16
امتیاز پارامتر هفتم	11	10	11	10	10	11	11	12	12	12	11	11
امتیاز پارامتر هشتم	4	4	4	4	5	4	6	4	4	4	6	6
نتیجه	8.27	8.44	10.2	8.13	9.13	9.3	11	7.16	7.54	7.97	8.51	7.47

در جدول شماره (۲) شماره رله‌های موجود اینگونه است که رله‌های روی شاخه اول به شماره‌های ۱ و ۱ هستند و رله‌های روی شاخه دوم به شماره‌های ۲ و ۲ هستند و... برای توضیح امتیازها در جدول (۲)، امتیازهای رله ۱ را شرح می‌دهیم. امتیاز پارامتر اول در جدول (۲) برای رله ۱ عدد ۳ است. مفهوم این عدد این است که رله ۱ در سومین ردۀ فاصله‌ای نسبت به دو منبع است. امتیاز پارامتر دوم عدد ۴ است. مفهوم این عدد این است که سطح اتصال کوتاه جلوی رله ۱، ۴ پریونیت در مبنای 1000MVA است. امتیاز پارامتر سوم عدد صفر است. مفهوم این عدد این است که رله ۱ عنصر سریع ندارد. امتیاز پارامتر چهارم عدد ۹۸ است. مفهوم این عدد این است که رله ۱، دارای ۹۳ رله هم حلقه است. امتیاز پارامتر پنجم عدد ۳ است. مفهوم این عدد این است که رله ۱، در حلقه‌ای مشترک است که تعداد رله‌های آن حلقه‌ها حداقل ۳ واحد کمتر از تعداد رله‌های حلقه با بیشترین تعداد رله است. امتیاز پارامتر ششم عدد ۱ است. مفهوم این عدد این است که در شاخه وصل به این رله حفاظت پایلوت داریم. امتیاز پارامتر هفتم عدد ۱۱ است. مفهوم این عدد این است که رله ۱، دارای ۱۱ رله پشتیبان در شینه‌های وصل به شینه رله است. امتیاز پارامتر هشتم عدد ۴ است. مفهوم این عدد این است که رله ۱ در ردۀ چهارم دور از بارهای مهم غیر قابل قطع می‌باشد.

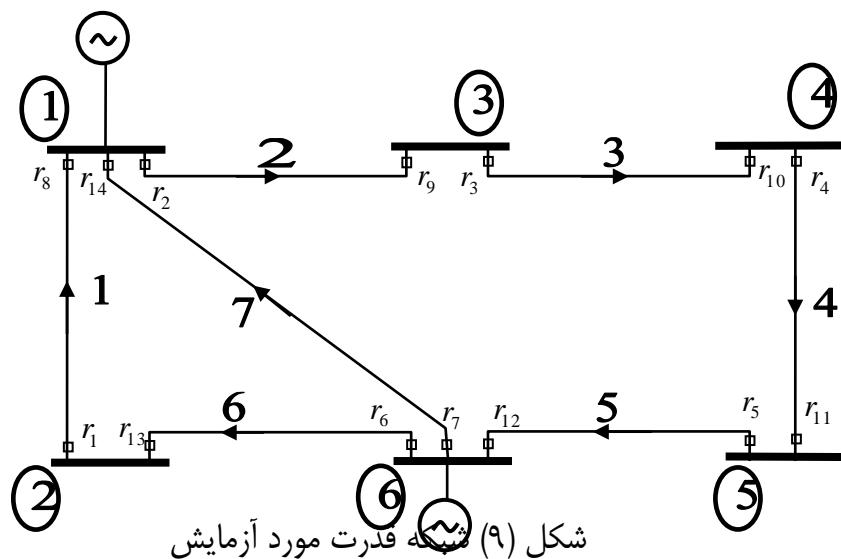
اگر بخواهیم از تئوری گراف استفاده کنیم، به ماتریس‌های *NRL* و *NRT* احتیاج داریم. این دو ماتریس سطرهای ۴ و ۵ جدول (۲) است آورده شده‌اند. در حالیکه در سیستم خبره بجای ۲ سطر از ۸ سطر برای سنجش استفاده شد.

در جدول (۲)، با مقایسه اعداد در سطر آخر این جدول، رله ۱۹ به عنوان بهترین نقطه شکست معرفی می‌گردد. چون بیشترین امتیاز را برای احراز نقطه شکست دارا می‌باشد. سپس با تغییرات لازم –که در بخش استنتاج عنوان شد و تکرار امتیازدهی مجموعه نقاط شکست با رله‌های ۱۹ و ۷ و ۱۵ و ۱۸ و ۱۰ و ۱۴ و ۹ و ۲۳ مشخص می‌گردد.

با استفاده از تئوری گراف برای حل این مسئله، نه تنها پارامترهای دیگری غیر از توپولوژی شبکه وارد نمی‌شد، بلکه در همان مرحله اول بین انتخاب رله‌هایی که در ماتریس‌های *NRL* و *NRT* امتیازهای مساوی دارند دچار انتخابهای متعدد می‌شدیم.

شبکه دوم مورد آزمایش:

شبکه دوم مورد آزمایش بصورت شکل (۹) است.



شبکه مطالعه شده در شماره ۱ یک شبکه نمونه بود. ولی این شبکه یک شبکه استاندارد است که اطلاعات آن به صورت زیر می‌باشد.

جدول (۳) اطلاعات خطوط و امپدانس‌های

شماره خط	شین انتهای	شین ابتدای	رله ابتدای خط	رله انتهای خط	راکانس (PU)	مقاومت (PU)
1	1	2	1	8	0.03333	0.00267
2	3	1	9	2	0.03332	0.00266
3	4	3	10	3	0.03002	0.00267
4	5	4	11	4	0.03	0.00333
5	6	5	12	5	0.03	0.0033
6	2	6	13	6	0.03	0.00264
7	1	6	14	7	0.03333	0.00333

اطلاعات مورد نیاز در بالا برای محاسبه جریان اتصال کوتاه جلوی هر رله است. حال به اطلاعات مورد نیاز برای اجرای برنامه پیدا کردن نقطه شکست می‌پردازیم.

جدول (۴) : اطلاعات سیستم قدرت

پیدا کردن نقاط شکست

۶۷

شماره شانخه	جهه ابتداء	جهه انتها	وصل به شیوه ابتداء	وصل به شیوه انتها	داشتن عذر سریع راه	داشتن عذر سریع راه	جهه ابتداء	جهه انتها	سطح اتصال کوئنده	پایلوت	داشتن محدودات	وصل بین شیوه ابتدایی منتهی	وصل بین شیوه انتهایی منتهی	همم بین شیوه از نظر پار
1	2	1	0	1	4.8	10.3	0	0	1	0				
2	1	3	1	0	11.3	2	0	1	0	0				
3	3	4	0	0	6.4	3.2	0	0	0	0				
4	4	5	0	0	6.2	7.7	0	0	0	0				
5	5	6	0	0	2.01	11.3	0	0	1	1				
6	6	2	0	1	10.3	4.85	0	1	0	0				
7	6	1	0	0	8.9	8.9	0	1	1	0				

این شبکه برای حالت بدون حفاظت پایلوت بررسی می‌گردد. در دو مثال، مثال‌ها با حفاظت پایلوت بررسی می‌گردند.

با ورود این اطلاعات به برنامه گراف، امتیاز‌های رله‌های شبکه برای پارامترهای متفاوت حساب می‌شود. این امتیازها در جدول (۵) آمده است. پس از وزن دادن به هر پارامتر حاصل نیز در جدول (۵) آمده است. در این مثال فرض بر این است که وزن تمام پارامترها با هم برابر است. یعنی ضریب وزنی برای هر کدام از پارامترها مساوی در نظر گرفته شده است.

جدول (۵) : امتیاز رله‌های شبکه

شماره رله	1	2	3	4	5	6	7	8	9	10	11	12	13	14
امتیاز پارامتر اول	2	1	2	0	0	2	1	1	0	0	2	1	1	1

پیدا کردن نقاط شکست

۶۸

امتیاز پارامتر دوم	4.8	11.3	6.4	6.2	2.01	10.3	8.9	10.3	2	3.2	7.7	11.3	4.85	8.9
امتیاز پارامتر سوم	1	0	0	0	0	0	0	1	0	0	0	0	0	0
امتیاز پارامتر چهارم	9	11	11	11	11	9	8	9	11	11	11	11	9	8
امتیاز پارامتر پنجم	0.16	0.16	0.16	0.16	0.16	0.16	0.2	0.16	0.16	0.16	0.16	0.16	0.16	0.2
امتیاز پارامتر ششم	0	0	0	0	0	0	0	0	0	0	0	0	0	0
امتیاز پارامتر هفتم	5	6	4	3	4	5	6	6	4	3	4	6	6	6
امتیاز پارامتر هشتم	3	3	3	2	1	3	3	3	3	2	1	2	2	2
نتیجه	5.9	5.33	5.6	3.54	3.01	5.39	5.01	6.06	3.6	3.2	4.5	5	4.2	4.6

در جدول (۵)، با مقایسه اعداد در سطر آخر این جدول، رله ۸ به عنوان بهترین نقطه شکست معرفی می‌گردد. چون بیشترین امتیاز را برای احراز نقطه شکست دارا می‌باشد. سپس با تغییرات لازم -که در بخش استنتاج عنوان شد و تکرار امتیازدهی مجموعه نقاط شکست با رله‌های ۱ و ۲ و ۷ و ۶ مشخص می‌گردد.

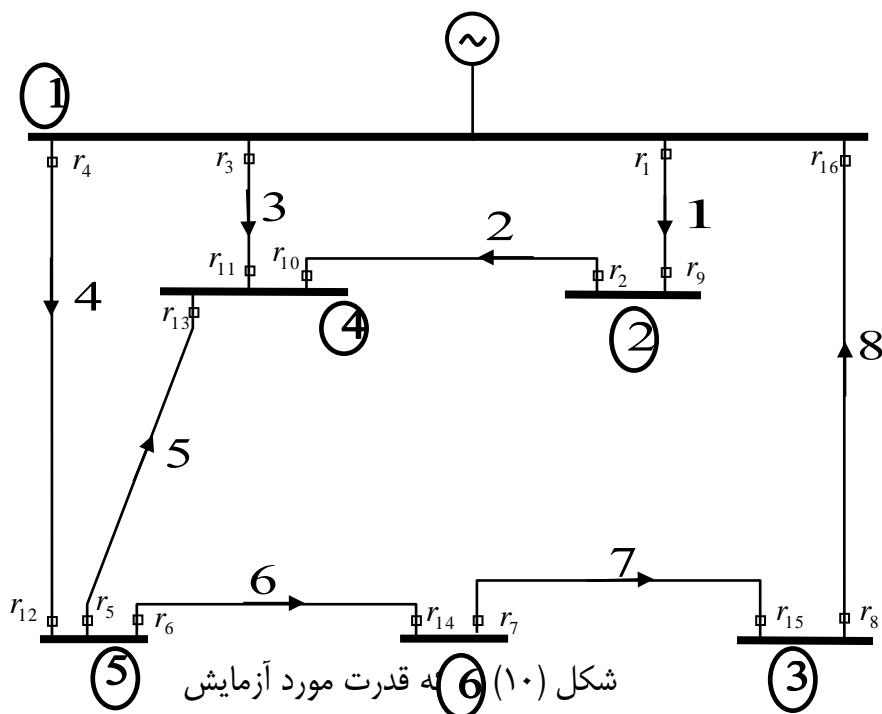
حال به تحلیل جواب بدست آمده می‌پردازیم:

می‌بینیم که رله‌های انتخاب شده در نزدیکی منبع واقع شده‌اند. این مسأله به چند دلیل درست است. اول اینکه سطح اتصال کوتاه در رله ۱ کم است. دوم اینکه عناصر سریع که در روی رله‌های ۱ و ۲ قرار دارند، باعث شده‌اند که این رله‌ها نقطه شکست انتخاب شوند.

حال می‌توان با یک تغییر وزنی برای دوری از منبع یا سطح اتصال کوتاه به جوابهای دیگری که معقولتر نیز هستند رسید. به فرض مثال ضریب وزنی سطح اتصال کوتاه و دوری از منبع را بجای مساوی با وزنهای دیگر، ۳ برابر بقیه گرفت. در این صورت به مجموع نقاط شکست با رله‌های ۶ و ۳ و ۱۱ می‌رسیم. از روی شکل می‌بینیم که این رله‌ها (رله‌های ۳ و ۱۱) دور از منبع و یا سطح اتصال کوتاه پایینتر قرار گرفته‌اند. ولی همچنان رله‌های ۶ و ۱ که در نزدیکی منبع قرار دارند نقطه شکست هستند. این مسأله بخاطر اینست که در هر صورت حلقه‌ای که با رله‌های ۸ و ۱۴ و ۷ و ۶ و ۱۳ و ۱ ساخته می‌شود باید از دو جهت باز شود. پس به اجبار دو رله از این مجموعه رله‌ها باید انتخاب گردد که این دو رله در این قسمت رله‌های ۶ و ۱ هستند. دیدیم که با تغییر ضرائب وزنی به جوابهای بهتری از دید هماهنگ کننده رله‌ها رسیدیم.

(۳-۴-۳) شبکه سوم مورد آزمایش:

شبکه چهارم مورد آزمایش بصورت شکل (۹) است.



اطلاعات شبکه بالا که قسمتی از شبکه انگلستان است. اطلاعات مورد نیاز برای امپدانسها از آن جهت که سطوح اتصال کوتاه را دهداند احتیاج نیست و فقط به اطلاعات مورد نیاز برنامه پیدا کردن نقاط شکست می‌پردازیم.

اطلاعات مورد نیاز در بالا برای محاسبه جریان اتصال کوتاه جلوی هر رله است. حال به اطلاعات مورد نیاز برای اجرای برنامه پیدا کردن نقطه شکست می‌پردازیم.

جدول (۶) : اطلاعات سیستم قدرت

پیدا کردن نقاط شکست

۷۰

شماره شانخه	مره ابتدا	گره انتها	داشتن عذر سریع ره	وصل به شنیده ابتدا	داشتن عذر سریع ره	جلوی راه ابتدا	سطح اتصال کوتاه	جلوی راه انتها	پایلوت	داشتن محدودات	وصل بیرون شنیده انتهاه منتهی	وصل بیرون شنیده ابتداه منتهی	همم بیرون شنیده از نظر بار
1	1	2	0	0	82.3	20.6	0	1	0	0	0	0	0
2	2	4	1	0	55.2	51.7	1	0	0	0	0	0	0
3	1	4	0	0	82.2	27.5	0	1	0	0	0	0	0
4	1	5	0	0	82.	21.6	0	1	0	0	0	0	0
5	5	4	0	1	50.4	70.1	0	0	0	0	0	0	0
6	5	6	0	0	53.4	40.3	1	1	0	0	0	0	0
7	6	3	0	0	18.3	50.3	0	0	0	0	0	0	0
8	3	1	0	0	12.6	82.2	0	0	0	0	0	0	0

این شبکه برای حالت بدون حفاظت پایلوت بررسی می‌گردد. در دو مثال، مثالها با حفاظت پایلوت بررسی می‌گردند.

با ورود این اطلاعات به برنامه گراف، امتیازهای رله‌های شبکه برای پارامترهای متفاوت حساب می‌شود. این امتیازها در جدول (۵) آمده است. پس از وزن دادن به هر پارامتر حاصل نیز در جدول (۵) آمده است. در این مثال فرض بر این است که وزن تمام پارامترها با هم برابر است. یعنی ضریب وزنی برای هر کدام از پارامترها مساوی در نظر گرفته شده است.

جدول (7) : امتیاز رله‌های شبکه

پیدا کردن نقاط شکست

۷۱

شماره رله	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶
امتیاز پارامتر اول	۱	۲	۱	۱	۲	۲	۰	۰	۰	۲	۰	۰	۲	۰	۲	۱
امتیاز پارامتر دوم	82.3	55.2	82.2	82	50.4	53.4	18.3	12.6	20.6	51.7	27.5	21.6	70.1	40.3	50.3	82.2
امتیاز پارامتر سوم	۱	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰
امتیاز پارامتر چهارم	13	13	11	11	18	15	15	15	13	13	11	11	18	15	15	15
امتیاز پارامتر پنجم	0.16	0.16	0.2	00.2	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.1	0.1	0.1
امتیاز پارامتر ششم	۰	۱	۰	۰	۰	۱	۰	۰	۰	۱	۰	۰	۰	۱	۰	۰
امتیاز پارامتر هفتم	۹	۶	۹	۹	۸	۸	۴	۵	۶	۸	۸	۸	۸	۴	۵	۹
امتیاز پارامتر هشتم	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
نتیجه	4.8	4.7	3.9	4.1	4.1	5.03	2.1	2.2	3.3	4.9	2.6	2.7	4.4	3.4	3.6	3.9

در جدول (۵)، با مقایسه اعداد در سطر آخر این جدول، رله ۶ به عنوان بهترین نقطه شکست معرفی می‌گردد. چون بیشترین امتیاز را برای احراز نقطه شکست دارا می‌باشد. سپس با تغییرات لازم -که در بخش استنتاج عنوان شد و تکرار امتیازدهی مجموعه نقاط شکست با رله‌های ۶ و ۱۰ و ۲ و ۵ و ۱۳ و ۱۵ مشخص می‌گردد.

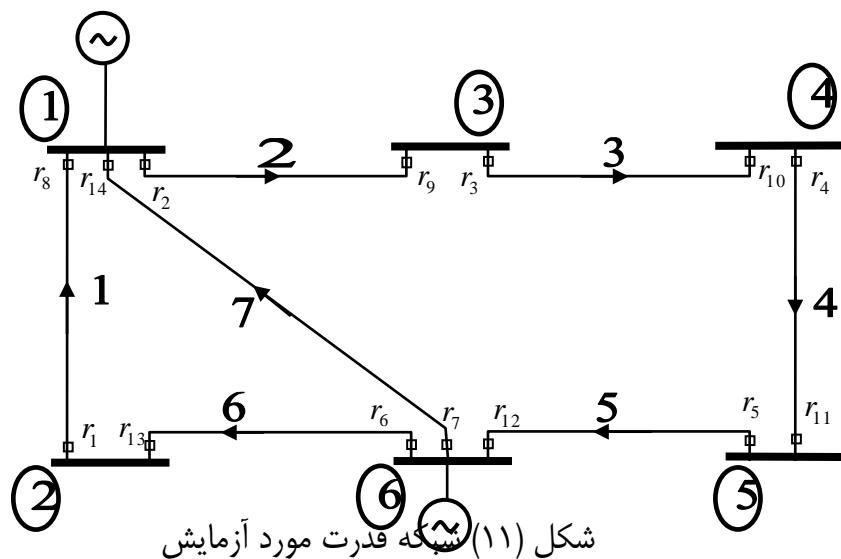
حال به تحلیل جواب بدست آمده می‌پردازیم:

می‌بینیم که رله‌های انتخاب شده دور از منبع یعنی باس ۱ هستند. همچنین حافظت پایلوت به خوبی استفاده شده است. یعنی رله‌ها طوری قرار گرفته‌اند که پس از هماهنگی اگر در آخر کار برای رله‌های ۲ و ۶ و ۱۰ و ۶ ناهمانگی یا هماهنگی بد اتفاق افتد، حافظتهای پایلوت می‌توانند این مشکل را پوشش داده و باعث جلوگیری از تغییرات در سیستم هماهنگی بشوند.

حلقه مستقلی که از رله‌های ۱ و ۹ و ۱۰ و ۱۱ و ۳ تشکیل شده است، چون یک حلقه است، باید با دو رله باز شود. این دو رله باید طوری باشند که حلقه را در دو جهت ساعتگرد و پاد ساعتگرد باز کنند. چه بهتر که این دو رله طوری انتخاب شوند که حافظت پایلوت موجود در روی خط شماره ۲ بتواند ناهمانگی‌های احتمالی را پوشش دهد. به این سبب دیده می‌شود که برنامه خبره رله‌های ۲ و ۱۰ را برای نقطه شکست پیشنهاد می‌کند.

(۳-۴-۴) شبکه چهارم مورد آزمایش:

شبکه چهارم مورد آزمایش بصورت شکل (۹) است.



شبکه مطالعه شده در شماره ۲ دارای حفاظت پایلوت نبود. حال شبکه مورد نظر را با حفاظت پایلوت در نظر می‌گیریم و برنامه را روی آن تست می‌کنیم و به تحلیل جوابها با وجود حفاظت پایلوت می‌پردازیم.

اطلاعات مورد نیاز برای شبکه و برای محاسبات اتصال کوتاه، همان اطلاعات مثال دوم است. ولی اطلاعات سیستم در زیر آمده است.

پیدا کردن نقاط شکست

۷۳

جدول (۸) : اطلاعات سیستم قدرت

شماره شانه	گره ابتدا	گره انتها	دایتین عصر سریع راه	وصل به شینه ابتدا	دایتین عصر سریع راه	سلخ اتصال ابتدا	جلوی ابتدا	سلخ اتصال کوتاه	جلوی راه انتها	پایلوت	دانشی خطاگذار	وصل بودن شینه	ابتداهه منبع	وصل بودن شینه	از نظر باز	جهنم بودن شینه
1	2	1	0	0	4.8	10.3	0	0	1	0						
2	1	3	0	0	11.3	2	1	1	0	0						
3	3	4	0	0	6.4	3.2	0	0	0	0						
4	4	5	0	0	6.2	7.7	0	0	0	0						
5	5	6	0	0	2.01	11.3	1	0	1	0						
6	6	2	0	0	10.3	4.85	0	1	0	0						
7	6	1	0	0	8.9	8.9	0	1	1	0						

با ورود این اطلاعات به برنامه گراف، امتیازهای شبکه برای پارامترهای متفاوت حساب می‌شود. این امتیازها در جدول (۵) آمده است. پس از وزن دادن به هر پارامتر حاصل نیز در جدول (۵) آمده است. در این مثال فرض بر این است که وزن تمام پارامترها با هم برابر است. یعنی ضریب وزنی برای هر کدام از پارامترها مساوی در نظر گرفته شده است.

جدول (۹) : امتیاز رله‌های شبکه

پیدا کردن نقاط شکست

۷۴

شماره رله	1	2	3	4	5	6	7	8	9	10	11	12	13	14
امتیاز پارامتر اول	2	1	2	0	0	2	1	1	0	0	2	1	1	1
امتیاز پارامتر دوم	4.8	11.3	6.4	6.2	2.01	10.3	8.9	10.3	2	3.2	7.7	11.3	4.85	8.9
امتیاز پارامتر سوم	0	0	0	0	0	0	0	0	0	0	0	0	0	0
امتیاز پارامتر چهارم	9	11	11	11	11	9	8	9	11	11	11	11	9	8
امتیاز پارامتر پنجم	0.16	0.16	0.16	0.16	0.16	0.16	0.2	0.16	0.16	0.16	0.16	0.16	0.16	0.2
امتیاز پارامتر ششم	0	1	0	0	1	0	0	0	1	0	0	1	0	0
امتیاز پارامتر هفتم	5	6	4	3	4	5	6	6	4	3	4	6	6	6
امتیاز پارامتر هشتم	0	0	0	0	0	0	0	0	0	0	0	0	0	0
نتیجه	3.6	5.3	4.06	2.08	3.6	4.3	4.01	4.06	3.6	2.6	4.1	5.3	3.5	4.01

در جدول (۵)، با مقایسه اعداد در سطر آخر این جدول، رله ۲ به عنوان بهترین نقطه شکست معرفی می‌گردد. چون بیشترین امتیاز را برای احراز نقطه شکست دارا می‌باشد. سپس با تغییرات لازم -که در بخش استنتاج عنوان شد و تکرار امتیازدهی مجموعه نقاط شکست با رله‌های ۲ و ۱۲ و ۶ و ۷ مشخص می‌گردد.

حال به تحلیل جواب بدست آمده می‌پردازیم:

می‌بینیم که رله انتخاب شده که شماره ۲ و ۱۲ و ۶ و ۷ هستند که تا حدودی وجود حفاظت پایلوت را در نظر گرفته‌اند یعنی این که در جلوی رله شماره ۲ حفاظت پایلوت قرار دارد این به این معنی است که پس از هماهنگی رله‌های شبکه‌اگر هماهنگی بین رله شماره ۲ و رله شماره ۳ را چک کنیم و به ناهمانگی و یا هماهنگی بد برخورد کنیم اشکال بزرگی متوجه هماهنگی نمی‌شود چون حفاظت پایلوت کار خود را انجام می‌دهد.

برای اینکه ثاثیر حفاظت پایلوت را بیشتر کنیم می‌توان ضربیب آن را در ضرائب وزنی بالاتر گرفت. مثلاً ۳ برابر ضرائب وزنی دیگر. در این صورت مجموعه نقاط شکست همان ۲ و ۱۲ و ۶ و ۷ خواهد بود. این مسئله بدان دلیل است که دو حفاظت پایلوت مربوط به یکحلقه هستند و با در نظر گرفتن یکی از آنها دیگر تأثیر دومی در نظر نمی‌آید.

۴- نتیجه‌گیری و پیشنهاد ادامه کار

۱- نتیجه‌گیری

در این پژوهه با اعمال سیستم خبره نشان داده شده تئوری گراف برای پیدا کردن نقاط شکست یک سیستم قدرت بمنظور هماهنگی رله‌های آن شبکه، جامع نمی‌باشد. زیرا این تئوری فقط به توپولوژی شبکه می‌پردازد. در حالیکه برای یافتن نقاط شروع هماهنگی و ترتیب آنها به پارامترهای دیگری نظیر دوری و نزدیکی به منبع و سطوح اتصال کوتاه و داشتن عنصر سریع و ... که در پژوهه به آنها پرداخته شد احتیاج است. با ارائه قوانین سیستم خبره بر مبنای پارامترهای یاد شده، روشی بمراتب جامعتر از تئوری گراف پیشنهاد شد و مزایای آن عنوان گردید. این قوانین با در نظر گرفتن خرائی وزنی تبدیل به امتیازدهی برای الوبیت دادن به نقاط شکست شد. نتایج قوانین بر روی نمونه‌ای از شبکه قدرت بهمراه رله‌های نصب روی آنها اعمال گردید و نتایج در خروجی در جداولی آورده شد. خروجی‌ها در مقایسه با روش‌های معمول نظیر تئوری گراف بیانگر موققتیت روش یاد شده در پژوهه است.

۳-۲-۴- پیشنهاد ادامه کار

از آنجا که این پروژه دارای ایده جدیدی در زمینه پیدا کردن نقطه شکست است، مشخص است که برای تکمیل شدن احتیاج به ادامه اینکار توسط محققان دیگری که این پروژه را دنبال می‌کنند دارد. از اینرو در ذیل پیشنهاداتی برای ادامه کار آورده می‌شود. این پیشنهادات به نوعی مکمل پروژه نیز هستند که باید توسط ادامه دهنگان اصلاح شود.

۳-۲-۱- اضافه کردن قوانین به صورت تجربی: (نبودن منبع آگاه)

در ابتدای انجام پروژه بر آن شدیم که قوانین تجربی تعیین نقاط شکست را از مقالات پیدا کنیم. ولی مقالاتی که در این زمینه وجود داشت بسیار محدود بودند و فقط به روش‌هایی برای ساده کردن تئوری گراف پرداخته بودند. پس این منبع عملاً منبع خوب و مفیدی نبود. پس از آن بر آن شدیم که از مهندسان رولیاژ اطلاعات کافی را بدست آوریم. ولی مهندسین رولیاژ، با تاسف بسیار اطلاعی از پیدا کردن نقاط شکست نداشتند و عملاً تنظیمات آنها بر اساس الگو و نظم خاصی نبود. پس به فکر درست کردن قوانین از اطلاعات خودمان افتادیم که یک سری قوانینی که در پروژه آمده است از این جمله است. در ادامه کار پیشنهاد می‌شود که با برنامه موجود، انواع شبکه‌ها آزمایش شوند و قوانین جدیدی برای سیستم خبره بدست آورند. هر چه تعداد قوانین بیشتر باشد، احتمال رسیدن به جواب بهینه بالاتر خواهد بود.

۳-۲-۳- ایجاد یک سیستم خبره واقعی

کار اصلی در این پروژه، درآوردن یکسری قوانین بود. ترجمه تئوری گراف به صورت دو قانون نیز از اهم کارهای این پروژه بود. ولی پس از ایجاد این قوانین با اعمال ضرایبی جواب را بدست آوردیم. کار بهتر و دقیق‌تر آن بود که امتیازهای هر رله از پارامترهای مختلف توسط یک شبکه عصبی Learn شود و یا توسط یک سیستم فازی پیاده شود و یا با یک پوسته سیستم خبره نوشته شود. البته اصلی‌ترین کار ساختن سیستم خبره قوانین آن است. ولی

در ادامه کار پیشنهاد میشود از سیستم عصبی یا فازی یا خبره برای گرفتن نتایج نهائی استفاده شود

۳-۲-۴- وارد کردن جفت جریانهای مختلف

در قسمت (۲-۲-۳) که قانون پایین بودن سطح اتصال کوتاه را بیان کردیم پیشنهاد بسیار جالبی وجود دارد. از آنجا که در برنامه‌های هماهنگی از روش هماهنگی به روش دو جفت جریان و چار جفت جریان و ... حساب می‌شود، می‌توان امتیاز برای نقطه شکست را فقط برای جریان اتصال کوتاه برای خطای جلوی رله حساب نکرد. بلکه برای تمام جفت جریانهای اول و دوم و ... حساب کرد. سپس این امتیازها را به رله‌های مربوطه بدهیم و بقیه کارها به همان منوال انجام شود. در این روش در حقیقت تا ۵ یا ۶ قانون به قوانین سیستم خبره می‌توان افزود.

۳-۲-۴- وارد کردن فیوز در شبکه

در قسمت (۳-۲-۳) که قانون عنصر سریع داشتن رله را عنوان کردیم می‌توان ورود فیوز را به پیدا کردن نقاط شکست وارد کرد. می‌دانیم در جایی که فیوز قرار بگیرد، سرعت بالایی داریم. پس احتمال نقطه شکست بودن بالاست و این مطلب در پروژه آمده است. ولی اگر در جایی دو فیوز داشتیم، آنوقت باید بین آنها فیوز اصلی و پشتیبان را تعیین کنید و بعد به آخرین فیوز امتیاز نقطه شکست بودن بدهیم. این مسئله می‌تواند در ادامه پروژه برنامه خبره را کاراتر سازد.

۳-۲-۴- بررسی نقطه ژرف در شبکه‌های رینگ‌بیه عنوان یکی از قوانین

یکی از پیشنهاداتی که در اضافه کردن قوانین مطرح شد، وارد کردن نقطه ژرف می‌باشد. در شبکه‌های رینگ، نقطه‌ای وجود دارد که پائینترین ولتاژ را دارا است. این نقطه می‌تواند توسط یک محاسبه پخش بار بدست آید. سپس رابطه بین این نقطه و نقطه شکست پیدا شده و امتیازی به هر رله بدهیم.

۴-۳-۵- امتحان کردن قوانین با هماهنگی رله‌ها

پس از اینکه نقطه شکست بدست آمد، باید به هماهنگی رله‌ها پرداخت و چک کرد که آیا با این نقاط شکست به جواب بهتری می‌رسیم. در حقیقت این قسمت کار به چک کردن قوانین مطرح شده می‌پردازد. از جهتی باید توجه شود که بعضی از قوانین دارای همپوشانی‌هایی هستند. مثل سطح اتصال کوتاه و دوری از منابع. پس باید مقدار خطای وارد شده در این قوانین بخاطر همپوشانی‌های موجود نیز مورد بررسی قرار گیرد.

مراجع-ω

- [1] DUMBORG, M.J., RAMASWAMI, A., VENKATA, S.S., and POSTFOROOSH, J.M., "Computer aided transmission protection system design, Part I: algorithms", *IEEE Trans.*, 1984, *PAS-103*, (1), pp. Si-59
- [2] DWARAKANATH, M.H., and NOWITZ, L., "An application of linear graph theory for coordination of directional overcurrent relays", *Electric power problem, the mathematical challenge*, SIAM, Washington, 1980, pp.104-114
- [3] RAO, V.V.l3., and RAO, K.S., "Computer aided coordination of directional relays: determination of break points", *IEEE Trans.*, 1988, *PWDR-3*, (2), pp.545-548
- [4] MADANI, S.M., "A new graph theoretical scheme for coordination of protection systems: determination of break point set", *EUT repoct 98-E-309, Department of Electrical Engineering, Eindhoven University of Technology, 1998, ISBN 90-6144-309-1*
- [5] PRASAD, V.C., and PRAKASA, K.S., and RAO, A., "Coordination of directional relayes without generating all circuits", *IEEE Trans.*, Vol.6, NO.2, April 1991
- [6] MADANI, S.M., and RIJAHO, H, "Protection co-ordination: determination of the braek point set", *IEE proc.Gener.Transm.Distrib.*, Vol.145, No 6, November 1998
- [7] 'کاربرد تئوری گراف در حفاظت شبکه های قدرت', دکتر علیمحمد رنجبر و مهندس محمد رضا رکوهی مجله برق، سال ۱۳۶۷
- [8] 'تعیین ترتیب هم‌آهنگی رله ها با ریز پردازندگان', دکتر علیمحمد رنجبر و مهندس محمد رضا رکوهی مجله برق، سال ۱۳۶۷